

2012.01

# PROGRAMMER

# 程序员

## 回顾与展望

年度十大人物、产品、技术、图书

22 IBM 韩忠恒  
云计算的另一种趋势  
软硬件协同设计

18 黑莓突围

63 提高面试代码质量三要素

66 硅谷产品管理大师 Marty Cagan  
不做只会编码的人

77 我眼中的2011年

80 Flash之死

84 逆势 衰退的开发者信心和  
不断示好的开发环境

102 Hadoop  
不是万能的

105 Beyond MapReduce  
2011年风靡的数据流计算系统

116 SDK安装程序卡壳之谜

125 云计算时代的多核开发

132 三好学生Chris Lattner  
LLVM 编译工具链

ISSN 1672-3252



邮发代号：2-665 定价：15元

www.programmer.com.cn www.csdn.net

# Contents



## 资讯

- 6 资讯  
8 新产品新工具  
10 外刊速递

## 热点报道

- 14 挑战下一个家伙  
——写在2011超级计算大会之后
- 16 性能追求，永无止境  
——Velocity China 2011观感
- 18 黑莓突围

## 企业专栏

- 13 三星智能电视应用开发简介
- 22 软硬件协同设计：云计算的另一种趋势

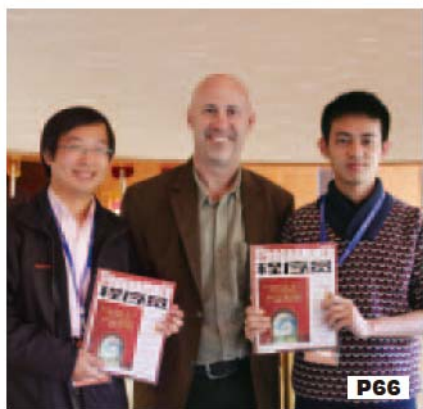


## 封面报道

2011年,整个IT业界发生了翻天覆地的变化,有太多的东西值得我们去回顾。展望2012年,我们有更多的期待。本期封面报道,采用TOP10的形式,全面聚焦热词、人物、产品、技术、图书和期待。

- |    |             |
|----|-------------|
| 24 | 年度十大热词      |
| 46 | 年度十大人物（国外篇） |
| 50 | 年度十大人物（国内篇） |
| 52 | 年度十大产品（国外篇） |
| 54 | 年度十大产品（国内篇） |
| 56 | 年度十大技术      |
| 58 | 年度十大图书（外版）  |
| 60 | 年度十大图书（本版）  |
| 62 | 2012年十大期待   |





## 管理

### 63 提高面试代码质量的三要素

作者总结自己多年面试他人以及被他人面试的经验，发现应聘者可以从代码的规范性、完整性和鲁棒性三个方面提高代码的质量。

### 66 不做只会编码的人

——硅谷顶级产品大师Marty Cagan专访

Marty Cagan是享有世界声誉的产品管理专家，曾担任Netscape副总裁、eBay产品管理及设计高级副总裁。近日，记者在“首届中国产品经理峰会”上对他做了专访，请他分享自己的产品管理历程。

### 70 产品经理的时间管理

### 74 敏捷团队中可视化的任务分配管理实践

本文分析了任务分配中的若干挑战与根本目标，并基于人力发展的一般模型，提出了一种可视化的任务分配深度管理之道。

### 77 我眼中的2011年

## 移动

### 80 Flash之死

Adobe公司决定停止面向移动浏览器的Flash开发工作，但导致Flash最终失败的原因却不甚明朗。VisionMobile特邀作者Francisco Kattan撰文就导致Flash消亡的一连串事件进行了分析。

### 84 逆势：衰退的开发者信心和不断示好的开发环境

目前，惨淡的收入似乎难以支撑开发者继续走下去的信心，然而本文作者却指出，开发环境正在改善。

### 87 《丁香园用药助手》成长之路

在移动化的道路上，丁香园团队遇到了哪些挑战与抉择？《程序员》将请他们分享面向垂直专业领域应用的成长之路。

### 89 用户体验专业团队的建设思路

用户体验团队的建设包括其组织形式、人员构成、成员招聘、业绩考核、团队管理等一系列学问。

### 93 移动应用开发技术选择六要素

作者从平台环境、操作系统、设备能力、云端、应用类型、跨平台开发六大方面分享了其在移动应用开发中的技术选择经验。

### 97 Mobile Web App发展现状及展望

作者指出，虽然性能上还存在很多缺陷，但从技术角度分析，Mobile Web App并没有太多致命门槛，正处于构建框架，重建规则的过程中。



主管：中国社会科学院  
主办：中国社会科学院文献信息中心  
出版：《程序员》杂志社  
网址：http://www.programmer.com.cn  
国际刊号：ISSN 1672-3252  
国内刊号：CN11-5038/G2  
邮发代号：2-665  
广告经营许可证号：京东工商广字0188号

总编：黄长著 Editor-in-chief: Huang Changzhu  
社长/常务副编：张悦校 President: Zhang Yuexiao  
副社长：蒋涛 Vice President: Jiang Tao  
编委会：黄长著 张悦校 陈洋彬 蒋涛 曾登高 刘江  
Editorial Member: Huang Changzhu Zhang Yuexiao Chen Yangbin  
Jiang Tao Zeng Denggao Liu Jiang

执行主编：孟迎春 Executive Editor-in-chief: Meng Yingxia  
编辑部主任：董世晓 Director: Dong Shixiao  
责任编辑：高松 陈博 陈秋歌  
Editors: Gao Song Chen Bo Chen Qiuge  
特邀编辑：方梁 高昂 赵健平 吕娜 卢鹤翔  
Contributing Editors: Fang Liang Gao Aang Zhao Jianping  
Lv Na Lu Dongxiang  
美术设计：纪明超 Art Designer: Ji Mingchao  
美术编辑：朱凯 Art Editor: Zhu Kai  
流程编辑：白羽中 Coordinator: Bai Yuzhong  
Tel: 010-64351458  
E-mail: editor@csdn.net

广告总代理：北京创新乐知信息技术有限公司  
Sole Advertising Agency: Beijing CSDN Co., Ltd  
Tel: 010-64376055  
E-mail: ad@csdn.net  
Marketing Dept: 010-51661202 (ext 149)  
E-mail: market@csdn.net

发行部  
Distribution Dept. 010-64351431  
E-mail: sales@csdn.net

读者服务部  
Readers service Dept.  
网上订购：http://dingyue.programmer.com.cn/  
读者信箱：reader@csdn.net

地址：北京市朝阳区广顺北大街33号院1号楼福码大厦B座12层  
Address: B-12th Floor Fairmont Tower NO.33 Guangshun North  
street, Chaoyang District, Beijing  
邮政编码：100102  
电话：010-64351436  
传真：010-64348545

法律顾问：北京中润律师事务所 王杰  
Law Consultant: Beijing Hengsheng Lawyer Firm  
印刷：北京盛通印刷股份有限公司  
Print: Beijing Shengtong Printing Co., Ltd.

出版日期：每月1日  
Publication Date: the first day per month  
零售价：RMB 15.00元 新台币 390元 HK \$ 35.00 (港、澳)  
US \$ 9.00 (海外)  
Retail Price: RMB 15, NT\$390, HK \$ 35.00, US \$ 9.00

本刊文章版权所有 未经许可不得转载  
发现装订错误或缺页，请将杂志寄回本刊读者服务部调换

## 云计算

### 102 Hadoop不是万能的

——Hadoop in China发起人谈Hadoop现状

2011年，对Hadoop来说注定是载入史册的一年，整个生态系统呈现出一派欣欣向荣的景象。近日，记者专门采访了Hadoop in China的两位发起人，为大家诠释Hadoop的现状。

### 105 Beyond MapReduce：谈2011年风靡的数据流计算系统

在MapReduce计算模型风靡全球之后，无论在工业界还是学术界，数据流计算都将会是下一个热点。本文从深层次对各种典型的数据流计算系统架构及其基于的设计理念进行了剖析。

## 技术

### 111 谈保险公司数据服务平台的建立

本文从作者负责在保险公司建立数据服务平台的实践出发，阐述了该保险公司建立数据服务平台的背景、选取的各种工具和工具的使用效果，以及建立数据服务平台的具体步骤。

### 114 前端技术这一年

前端工程师的2011年惊喜不断。很多前端技术如同寒武纪大爆发，得到了长足的发展与丰富，本文从国外、国内两个角度，回顾了前端技术的热点话题。

### 116 SDK安装程序卡壳之谜——兼谈函数的异常出口（下）

书接上回，本期继续探索Windows 7 SDK安装程序卡壳之谜，彻底还原文件下载线程“误入罗网”的来龙去脉。

### 120 深入Zend线程安全模型

本文首先解释了线程安全的概念及PHP中线程安全的背景，然后详细研究了PHP的线程安全机制ZTS及具体实现，最后探讨了Zend对于单线程和多线程环境的选择性编译问题。

### 125 云计算时代的多核开发

云计算时代的多核开发会是一幅什么样的场景？这两大趋势彼此会有什么样的影响？本文就这些问题做了探讨。

## 程序春秋

### 132 三好学生Chris Lattner的LLVM编译工具链

12月3日，LLVM 3.0正式版发布，完整支持所有ISO C++标准和大部分C++ 0x的新特性。这对于一个全新项目来说非常不易。

### 128 新书上架

### 130 Geek

### 136 程序幽默

# 变与不变——2012 新年献辞



如果用一字来形容2011年的话，我会继续选择“变”。

随着移动和云计算前后两端的兴起，无论是技术、平台还是企业，整个IT行业诸多层面的变迁用沧海桑田来形容也不为过。

2011年，我们看到Objective-C、Lua因移动而在编程排行榜上快速提升；Java的霸主地位依旧，而Scala等JVM语言也得到实际生产环境的检验；JavaScript更是渐渐显示出统一前后端的潜力，Harmony、CoffeeScript、Dart等下一代JavaScript，与Node.js、jQuery等众多第三方框架和库组成了生机勃勃的生态系统。

2011年，我们乐见敏捷进入主流，不仅在更多国内各类规模、行业的软件团队中发挥着重要作用，也开始辐射到嵌入式软件、用户体验、组织变革和企业文化等诸多新领域。

2011年，我们目睹“数据中心成为新的计算机”这一云计算时代的大趋势渐成共识，多核/众核芯片、ARM架构开始成为主流硬件平台，SSD和Fabric计算、低功耗定制服务器、内存计算、异构等诸多较新技术都得到更多关注和应用。而软件领域，Hadoop、NoSQL等大数据技术和DevOps、持续交付等运维实践更是火爆，并获得了技术社区的广泛支持。

2011年，我们见证了Google、Apple、Facebook、Amazon、腾讯、阿里、百度、新浪等互联网的迅猛发展，并通过切入点各异的开放平台、应用商店构建各自的生态链，而摩托罗拉、诺基亚、惠普、索尼、黑莓等等曾经的行业领先者难以赶上变化的步伐。Flash、Silverlight、Win32几大主流平台性技术走向衰落，Wintel联盟的统治地位被ARM+Android/iOS全新组合冲得阵脚大乱。

2011年，我们经历了Zynga、Groupon、Jive、PopCap，和愤怒的小鸟、水果忍者、二战风云、捕鱼达人、海豚浏览器等等新生力量或者成功上市、被并购结成正果，或者高歌猛进得到数以亿计的用户和收入，或者获取了千万美元计的投资。他们的故事，激励着更多后继者走上创业之路。当然泡沫也随之而起，薪资暴涨，人心浮躁，概念、跟风 and 快速抄袭成了时尚。

2011年，我们还送走了几位本时代最伟大的同行：给人诸多启发的Steve Jobs，还有Dennis Ritchie和John McCarthy，分别作为C、Unix、开放系统与Lisp、人工智能、分时系统的缔造者，没有他们的贡献，今天的计算机系统可以说根本就不可能存在。

那么，2012年呢？

永恒不变的只有变化，全球经济的不确定性必将对IT行业产生影响，但历史上这种看似不利的局面往往成为大发展的前因。一些长期不变的真知灼见比预测更有意义。

新的一年中更多变革，可能来自互联网技术和模式对各个传统行业的渗透和颠覆。2011年Mosaic浏览器开发者、目前硅谷最重要的投资人之一Marc Andreessen在*Why Software Is Eating The World*一文中直言，随着宽带和智能移动设备的普及，IT基础设施成本降低，诸多创新型的软件和在线服务正在掀起一场革命，逐一颠覆传统行业：音乐、电信、媒体出版、电视、零售、招聘、石化、农业、金融、医疗和教育、国防……怎样应对这样的变革？传统行业必须争分夺秒地互联网化、软件化，重视技术和研发。创业者和创新公司应该从需求出发，在万事万物数字化、网络化和分析化中，寻找通过用互联网和软件技术解决现实问题的各种机会。

而在乔布斯的思想遗产中，我认为宏观层面最有价值的一点也在于，Apple整个公司都以产品研发和研发人员为中心组织和优化，吸引最优秀的研发人员（工程师和设计师），不断为用户推出更好的产品，是公司的第一要务，甚至不惜牺牲正在挣钱的现金牛产品和短期的利润，实际上在不断颠覆自己，从而打破了Christensen的“创新者窘境”魔咒。

优秀的研发人员已经成为整个生态链的重中之重。Amazon公司的HR @小F小Fairy的一条微博让我大受感动，她这样总结优秀程序员的共性：

谦称自己为程序猿、码农、民工……；

非常低调，喜欢自嘲；

愤青，但很关心社会和民生问题；

迷恋美女；

不容易沟通，但是一旦碰对了就会交心，而且交的很深。

是的，我们的研发人员是当代最可爱的群体之一。他们无需背景，不用拼爹，仅仅用自己的热情、脑力和技术构建社会赖以运转的基础，默默地推动着世界向前发展。

新年到了，祝天下所有好产品都有好回报，祝天下所有做好产品、写好代码的人阖家幸福。

邮件：liujiang@csdn.net

新浪微博：@刘江CE

欢迎大家交流反馈，欢迎投稿、批评、建议和挑错



## 云计算

根据IDC的报告,在未来十年,全球大数据将增加50倍。仅在2011年,就有1.8ZB(也就是1.8万亿GB)的大数据创建产生,相当于每位美国人每分钟写3条推文,不停写上2.6976万年。在未来十年,管理数据仓库的服务器数量将增加10倍,才能应对50倍的大数据增长。这对拥有大量数据的互联网企业提出了急迫的挑战,数据基础设施建设成为当务之急。其数据中心往往需要运行海量的云计算及流量处理。于是各大企业纷纷在海外建立自己的数据中心。

Facebook已经与瑞典企业合作在北极地区建立数据中心,以有效利用当地的可再生能源,当地寒冷的气候也将有助于冷却其服务器设备。Google也将投资1.2亿美元在新加坡建设数据中心,这是该公司在东南亚建设的第一个数据中心。日立还将在中国建立两个新的数据中心,以推动中国业务的发展。

毫无疑问,大数据将对企业的存储架构及数据中心基础设施等提出挑战,也会引发云计算、数据仓库、数据挖掘和商业智能等应用的连锁反应。2012年企业会将更多的多TB数据集用于商务智能和数据分析;到2020年,全球数据使用量预计暴增44倍,达到35.2ZB。

## iPad改名

苹果iPad在华商标案一审败诉,知情人士透露,有国内企业正寻求获得iPad的中国商标权,以便生产相关产品。2011年12月,纠缠两年之久的iPad商标案在深圳中级人民法院作出一审判决,苹果公司要求确认iPad在华商标归属权的请求被驳回。这一判决意味着,苹果公司并不拥有iPad在中国内地的商标权,不过苹果还有权就此判决提出上诉。

据悉,iPad在中国内地的商标权归属深圳唯冠公司,但由于金融危机,深圳唯冠几近破产,所有资产都已经被中国银行、民生银行等8家银行接管。受银行的委托,和君创业总裁李肃担任了深圳唯冠的债务重组顾问。他表示,苹果公司在这件事情上没有赢的希望,处于绝对被动地位,苹果公司一定会提起上诉,以便拖延判决生效时间。

据法律专家解读,一旦最终判决深圳唯冠拥有iPad在中国内地的商标权,深圳唯冠则可以对苹果公司提前追溯侵权行为,并索要赔偿,但由于判定商标侵权损失过程复杂,双方最终可能会达成和解。



## 移动

尼尔森发表报告展示了2009-2011年移动市场普及状况:美国智能手机普及率从18%暴增到44%。iPhone成为最流行的单一终端,Android市场占有率最高,为智能手机总销量的53%。18-34岁之间的人群有超过一半在使用智能机,大龄群体的使用率也增加了1倍以上。

如此迅猛的发展速度,不可避免地暴露了一些弊端。以Android为例,在受用户青睐的同时,恶意软件也随之增加。据研究显示,美国用户遇到恶意软件的概率为4%,而其他诸如不安全链接等安全隐患则更为常见,40%的美国Android用户可能会遇到。

因为苹果的iOS以及微软的Windows Phone平台对于应用上线都有相对严格的审核机制,所以恶意软件的重灾区主要在于Google的Android系统。在2011年上半年,大约有100万用户被Android恶意软件感染,受影响应用程序超过400个。移动恶意软件在4个月中以惊人的800%的速度增长。

2011年,全美Android用户因移动安全被盗的总金额约为100万美元,2012年可能会更多。僵尸网络的阴影迟早会笼罩移动领域。监控手机操作数据的案例时有发生,Carrier IQ以及HTC和三星两家手机厂商就因此遭到美国用户的集体诉讼。

值得警觉的是,国内移动的发展可能更为迅速,安全问题可能更为严峻。

“再次重申,Kindle Fire将成为主流平板,关键原因是价格有吸引力。”

——麦格理证券分析师Ben Schachter认为虽然Amazon没提供更具体的销售数据,但初步数据足以说明,之前定下的销售目标偏保守,因此他将Amazon股票评级为“买入”。

“我们的方法更像是星际迷航中的进取号星舰,每一个操作都基于语音识别,没有任何人的因素,它也没有任何名字,它只是计算机。”

——Google正在研发一款对抗苹果Siri的Android应用,代号Majel。对此Google高管Matias Duarte曾经如此暗示说。

“令人畏惧的强大的谈判者。”

——Google董事长Eric Schmidt此前如是说。Zynga更新了IPO招股书,列出了失去CEO Mark Pincus的风险——

“Pincus对我们的目标、战略方向、文化、产品和技术非常重要。失去我们的创始人和CEO,即使是临时性的,或是失去高管团队的其他成员,都将对我们的业务造成损害。”

“广告注入是一种不公平且带有欺诈性和迷惑性的行为。”

——Facebook将推出一种技术方案,封杀那些干扰其广告业务并引发用户不满的软件。这类软件会在网页上添加额外的广告层,同时屏蔽部分内容和Facebook出售的广告。Facebook法律事务主管Craig Clark对此事做如上评价,并表示在Facebook收到的广告投诉中,有80%~90%都来自这类应用发布的广告。

“我们认为一般用途的处理器无法与像A5这样的专用功能SoC(片上系统)处理器竞争。”

——美国投资银行Piper Jaffray高级研究分析师Gus Richard发布报告称,苹果已经在移动芯片领域大幅领先Intel。

“此时此刻我想说,如果这是一种威胁的话,那我希望这将会引起立法者积极的思考。不要低估我们的力量,在我看来,公众对于这件事的意见非常大,我们时刻准备着。”

——美国为打击在线盗版现象而草拟SOPA法案,这被看做是美国第一个互联网审查制度。对此,维基百科联合创始人Jimmy Wales表示维基百科网站应该临时关闭,以表达对该法案的抗议。

新浪微博用户每天发8000万条微博,45%来自手机,其中,iPhone用户占36%,Android占25%,而Symbian则一直在下降,其他平台可以忽略不计。

——新浪微博开放平台负责人芦义在2011年11月30日举行的“微博游戏开发者论坛”上如是说。



## CSDN十大资讯

2011年11月26日-12月26日

### 01 开发者为什么需要Mac?

作者 / Tim Anderson

作者指出开发者都应尝试开发基于Mac的应用，并列出四点理由。一、开发和测试基于苹果产品应用的需要；二、苹果用户增多；三、Mac现在占据很大优势，仅仅精通Windows开发将不再有优势；四、使用Mac可以避免糟糕的产品可用性。

### 02 雅虎公司，十年几无任何建树

作者 / Charles Arthur

为何没有人想要收购雅虎了？因为在过去的十年当中，雅虎没有做任何事情来制止公司的陨落。雅虎高层在掌握产业变化上一直存有缺陷。与雅虎形成对比的是，Google从来都不满足，而雅虎和杨致远却一直逍遥自在。这就是为什么雅虎走向衰败的根源。

### 03 微软可远程删除Windows 8应用程序

作者 / 彪赫

微软将在Windows Store中设置“Kill Switch”功能。该功能将远程禁用或删除用户设备上的应用。苹果的App Store和Google的Android Market都有这项功能。微软明确表示，公司将谨慎使用Windows Store中的“Kill Switch”功能，如果删除应用程序，可能给予用户原价赔偿。

### 04 Twitter内部到底出了什么问题

作者 / Nicholas Carlson

Twitter发展得顺风顺水，员工、用户和企业主数量都达到新高度。然而公司内部一直不太平，据一位匿名的前员工透露，Twitter的问题主要出文化上：办公室里充满负面情绪，工程师文化不足，没有达到比较优秀的技术公司50%以上是工程师的水平。

### 05 盖茨解释乔布斯讨厌微软原因

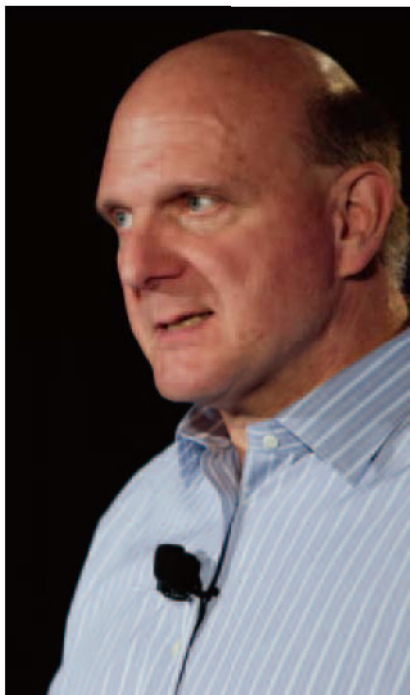
作者 / Fairfax Media

Bill Gates在接受澳洲媒体采访时表示，Jobs之所以对微软态度强硬，是因为“微软卖出的机器比苹果多得多”。同时，Gates表示认同Jobs是一个非常杰出的人，两人一起工作很享受。Gates还否认了重回微软的传言，表示全部精力将放在The Bill & Melinda Gates基金会上。

### 06 鲍尔默被评第三差CEO

作者 / Glassdoor.com

Glassdoor.com发布了美国50多家大型科技公司CEO的员工满意度排行榜。这是根据公司员工在其网站上发表的对CEO的看法编辑的。其中，微软CEO Steve Ballmer是第三差CEO，倒数第二位是雅虎的Timothy Morse。垫底的是施乐的Ursula Burns。



### 07 谷歌宣布正式关闭笔记本

作者 / 魏力

Google关闭笔记本产品，并将数据移至Google文档，称“为了对那些从长远来说能为用户带来最大收益的产品和技术投入更多精力，不得不做出一些艰难的决定。”Google宣布将结束、停止研发或限制存取6项服务，这些产品被Google认为对自身品牌或财务贡献不大。

### 08 Firefox资金状况存忧虑

作者 / Christopher Williams

Firefox市场占有率不断下滑，Mozilla资金状况令人担忧。Chrome浏览器对Firefox构成威胁。Google大力宣传Chrome的速度优势，并通过搜索引擎吸引用户使用Chrome。Mozilla有84%的收入来自与Google相关的业务，但两者的合作协议在2011年11月结束，还未达成续约。

### 09 小米手机被联通定制200万台

作者 / 侯继勇

小米手机已被中国联通选为定制手机，首批定制200万台，补贴方式或与iPhone 4保持一致。业内人士一致认为这是小米手机的重大利好，至少说明“稳定可靠”的订单问题已经得到解决。但能否解决生产体系、供应链体系才是小米手机成功的最大考验。

### 10 微软即将推出IE的自动升级功能

作者 / Ryan Gavin

据微软IE业务和营销总经理Ryan Gavin表示，微软将推出IE自动升级功能。“（微软）将于2012年1月，在已开启Windows自动更新的澳大利亚和巴西用户处最先进行尝试。”推出IE自动升级或许是希望用户不再使用IE6。IE6是目前浏览器市场最明显的一个妨碍。

## 2012年10大热门IT技术预测

EE Times网站推出一篇由3位作者Peter Clarke、Nicolas Mokhoff和Rick Merritt联合打造的2012年20大热门技术文章*EE Times 20 hot technologies for 2012*，文中评选出在未来一年中极有可能成为热点的20项技术。其中有10项是关于IT领域的，这些技术是否会如他们所愿被大家所推崇，还是会因为其他因素而没落？在时间的见证下，让我们一同期待2013年的到来。

### 01 Wireless Sensor Networks

无线传感网络是由许多分布式的自动装置组成的无线通讯计算机网络。使用传感器协作监控不同位置的物理或环境状况。

### 02 Internet of Things

物联网是基于互联网、传统电信网等信息承载体，让所有能够被独立寻址的普通物理对象实现互联互通的网络。

### 03 NFC

近场通信技术作为非接触式识别和互联技术，可以用于移动设备、消费类电子产品、PC和智能控件工具间的近距离无线通信。

### 04 Next-generation Non-volatile Memory

下一代非易失性存储器将使得存储元件在微缩能力、读取电流或是相互干扰等问题上，都获得改善。

### 05 Processors

处理器必须完成从multicore到manycore的转变，还必须应付新编程语言和环境的变化以及满足更低的能耗需求。

### 06 Graphics and GPGPU

通用图形处理器，可以做到更多的效果，不仅是呈现2D和3D效果。利用GPGPU可以并行处理、设计和渲染。

### 07 White Space Radio

频谱白空间利用数字电视和电台广播的可用频谱，还预示着一个提供机器间（M2M）通信的潜在平台的出现。

### 08 LTE

长期演进技术也被通俗地称为3.9G。LTE将极大地减轻移动数据网络的阻塞情况。LTE将向全IP网络迈出一大步。

### 09 40/100 Gbit/second Ethernet

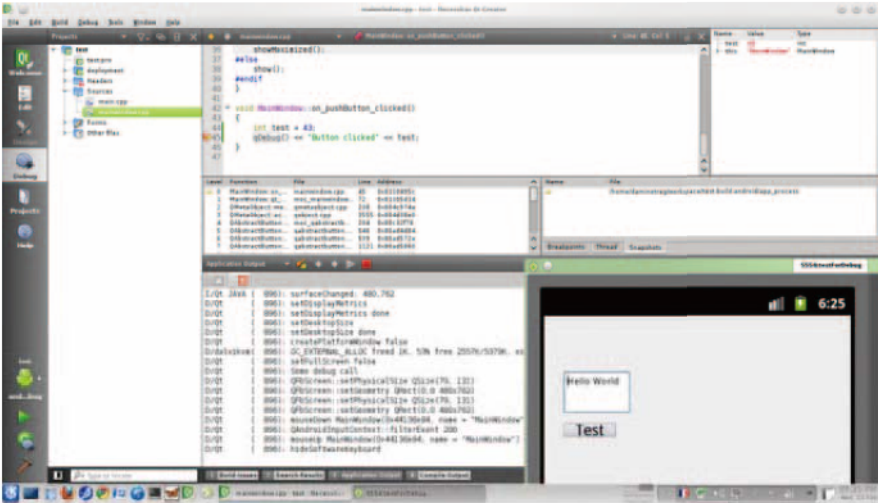
40/100 Gbit以太网是未来有线互联网世界将发生的一件大事，将会推动创新的光学技术，以降低成本和空间要求。

### 10 Mobile OSes with Android

在未来，这个绿色机器人将不光具有NFC功能，还将装备上现实增强技术、手机识别接口以及对HTML5更好的支持等。

## Android平台的Qt实现——Necessitas

Necessitas是一个用于Android平台的Qt开发实现。众所周知，Qt是诺基亚拥有的跨平台的C++图形用户界面应用程序框架。它提供给应用程序开发者建立艺术级的图形用户界面所需的所有功能。Qt是完全面向对象的，很容易扩展，并且允许真正的组件编程。Qt目前已经支持Windows、Linux和Mac OS X等平台，并且在智能手机操作系统领域，支持Windows Mobile、Symbian和MeeGo平台。目前Android平台在智能手机领域已经举足轻重，而Android目前只支持Java开发。于是便有了将Qt与Android相结合的Necessitas项目。



## Objective-C开发工具包ytoolkit

ytoolkit是Objective-C的一个常用开发工具包，包括：快速的base64编码；Cocoa附加库；完善的OAuth1库及相对应的NSMutableURLRequest和ASIHTTPRequest的扩展、OAuth2库及OAuth2 HTTP-MAC的扩展的实现。

## Grails 2.0正式版发布

Grails是一个全栈框架，它借助核心技术与相关的插件来解决Web开发中方方面面的问题，它基于Groovy和Java编程语言，并构建于Spring、Hibernate和其他标准的Java框架之上，利用了Java EE Sphere最好的API，为开发者带来一套能实现超高生产力的一站式框架。最新2.0版本从命令行，到测试报告，再到用户界面等方面都增强了用户体验。



## Lua 的Web工具包LWT

LWT（Lua Web Tools）可让你使用Lua开发Web应用，并可直接在Apache上运行。LWT的核心功能通过一个Apache的模块（mod\_lwt）来提供的。另外，LWT提供可选的Lua模块用来访问数据库和缓存。

## jQuery插件

jQuery是一个非常优秀的JavaScript框架，而且还有相当多的插件，在工作中使用能起到事半功倍的效果。Ninja UI是一款界面插件，包含有幻灯、智能提示等常见的效果；jCanvas是一个的绘图插件，它封装了一些绘制图形的方法，只需编写几行代码即可生成图形；jqPagination是分页插件，使用了HTML5和CSS3技术来实现。此插件提供了几个参数设置选项，通过简单的配置即可生成分页控件。此外，它的外观样式是可自定义的，扩展性很强。jBox是多功能对话框插件，可用鼠标拖动对话框、改变对话框大小等，能够实现网站的整体风格效果。



## 全新HTML5框架Amber.js发布

Amber.js的前身是SproutCore 2.0，是一个HTML5的Web应用框架，目标是在浏览器中提供桌面效果应用而无需任何浏览器插件。苹果的MobileME（现在的iCloud）即曾使用该框架。



## 移动Web开发框架LungoJS

LungoJS是一个使用HTML5、CSS3和JavaScript技术的移动Web开发框架。可帮助创建基于iOS、Android、BlackBerry和webOS平台的应用程序。该框架可利用当前移动设备的高级特性，可捕捉事件包括滑动、触屏和长按等。无须使用图片，全部采用向量声称。LungoJS无须服务器端支持，可帮助实现HTML5特性，包括WebSQL、Geolocation、History和Device orientation等。这是完全可定制的框架，可用来创建应用、游戏等程序。



## Facebook发布全新JIT PHP编译器

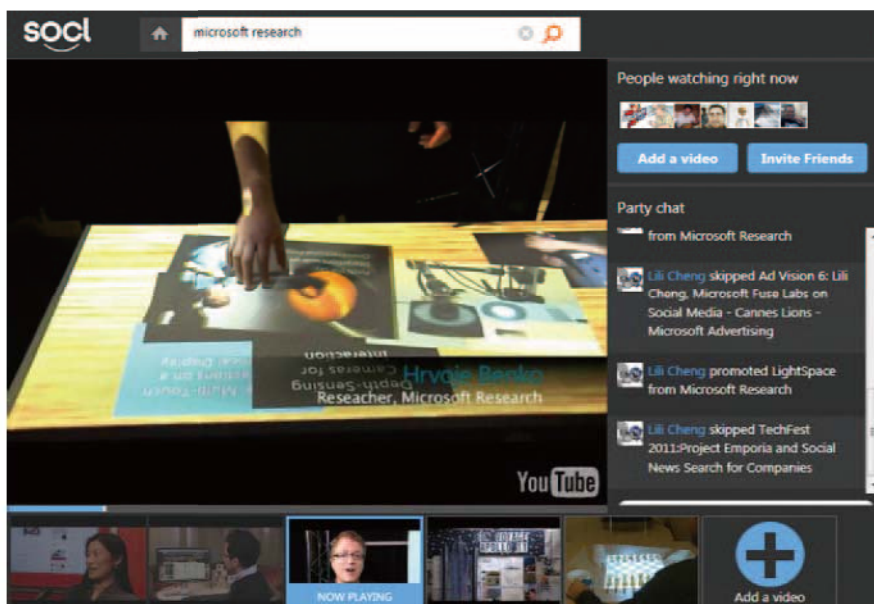
Facebook发布了名为HipHop虚拟机的编译器，可作为源代码转换器，将PHP代码转换为高度优化的C++代码，然后再使用G++编译器编译。HipHop被Facebook广泛运用，其网站所有PHP代码都经HipHop转换过，旗下其他网站也采用HipHop来优化。



## 微软So.cl正式发布

这是一项面向学生的社交服务，由微软FUSE实验室开发。So.cl提供与Facebook类似的功能。用户除了可以添加状态更新、照片和链接外，还可以加入“Video Party”，与该群组的好友一起观看电影、聊天。

该项目目前处于实验阶段，主要为就读于华盛顿大学、纽约大学和锡拉丘兹大学的学生提供社交服务。但微软在So.cl项目的FAQ页面称，So.cl并不是要取代目前功能完善的搜索、社交网络，这是一个实验性的研究项目，使用了最少的功能，主要用于探索社交网络和社交搜索对于学生学习的作用，并鼓励学生之间进行互动和协作。



## 基础架构云快速部署解决方案套件SKC发布

SKC是构建在PowerVM与VMControl之上的云计算管理平台软件，目前支持Power服务器和x86服务器。同时，还有三个主要功能：虚拟化管理，包括虚拟机的创建、运行与销毁，虚拟化资源池，对资源进行动态和可扩展的调度；自动化管理，提供用户自助服务、自动计量与计费、自动化虚拟机和影像管理等；标准化管理，将资源管理流程标准化，并对其生命周期管理实现标准化等。

## Palantir发布开源应用

Palantir发布的两款开源应用分别是Cinch和SYSMON。Cinch是简化Swing GUI开发的Java库，用于简化Swing MVC应用程序开发。它通过使用Java Annotation来实现MVC架构。SYSMON是一个轻量级的平台监视工具，用于收集Java虚拟机运行时的性能数据（CPU、磁盘占用和网络等）。收集的数据可由Java管理扩展（JMX）封装并发布，借助JMX API及其他工具（如jconsole）可以访问。



## Amazon——不仅仅是书店

2011年9月，Amazon宣布了新的Kindle产品系列，其中包括仅售79美元的Kindle Touch，以及我们期待已久的Amazon平板——Kindle Fire。199美元的价格，易于掌握的尺寸，与Amazon丰富且不断增加的数字内容无缝连接，都让业界为之兴奋。分析家将之誉为iPad的最强劲对手，但这款产品的诞生不是为了与iPad争锋，而是有本质上的不同。iPad是后PC时代的领航者——用小巧、便携和手势控制的平板代替桌面PC。而11月发布的Kindle Fire则是后互联网时代的象征，通过它，我们可以直接接入互联网上的数据中心。

虽然iPad和Kindle Fire的用途大致相同——看电影、读书和玩游戏，但隐藏在这两款平板产品背后的哲学是大相径庭的。基本上来说，苹果是一家硬件公司——其收益的91%来自于硬件产品的销售，相比之下iTunes的收益只占总量的6%；而Amazon则是一家以内容为中心的公司——其收益的一半来源于媒体内容的销售，包括图书、音乐、电视节目还有电影，而Kindle Fire的主要功能是作为通向这些海量数字内容的入口。Kindle

## Wired

2011.12

Fire预装了消费者的Amazon账户信息，如果你订购了年费为79美元的Amazon Prime服务，就可以通过Kindle Fire获取多达12000部的电影和电视节目，而不需任何额外费用。实际上，Amazon的CEO Jeff Bezos更倾向于将Kindle Fire称为媒体服务平台。Kindle Fire的发布展示了Bezos更长远的计划。人们会渐渐发现，Amazon的云服务在很大程度上推动了互联网的发展，渗透到我们生活的每个角落。Bezos很可能像Bill Gates和Steve Jobs一样成为互联网的传奇。

## 比特币的起落史

自互联网诞生以来，电子货币的构想就是个热门话题。它方便、不可追踪，而且能免于政府和银行的监管。20世纪90年代自由密码运动中诞生的“密码朋克”们，曾经致力于电子货币的研究，包括Ecash、Bitgold、RPOW和b-money，但是所有努力都以失败告终。

直到2008年11月，有人以Satoshi Nakamoto的名义在一个密码学讨论组上，发布了一篇研究报告，描述了他设计的一种新电子货币——比特币（BitCoin）。Satoshi Nakamoto的身份至今还是个谜，但是他的工作破解了困扰密码工作者多年的难题。

电子货币设计的核心挑战，是重复支付问题。如果电子货币仅仅是一段信息，而不受纸张或金属等物质形式的制约，那么如何防止人们像复制粘贴文本一样复制货币，随心所欲地花费呢？传统的解决方法是建立中央票据交换所，记

录每一笔交易的情况。这可以有效防止欺诈行为，但需要第三方机构的介入管理。比特币使得电子货币监管摆脱了对第三方机构的依赖，这种机制称作“区域链”。用户贡献自己的CPU资源，运行特殊的软件来“挖矿”，共同形成一个网络来维护“区域链”。在这个过程中，也会产生新的货币。比特币不依赖于政治家和金融家的可信任度，仅仅依赖于Nakamoto优秀的算法。它不仅能预防欺诈，而且其总量的增长在可预见范围内，能够避免无限度印发货币而造成的通胀。

2009年到2010年年初之间，比特币还没有什么实际的价值。2010年春天比特币交易开始。到2011年2月，比特币和美元的兑换率达到了1:1。由于价格的飙升，“挖矿”活动变得越来越热门。而在这一片繁荣中，危机也开始孕育。有人将比特币交易视为投机行为，呼吁禁止其流通。比特币的存储也是问题，数据的丢失让不少人手中的比特币损失惨重。而针对比特币的黑客行为也开始出现。

虽然比特币还有许多狂热拥护者。人们对比特币的质疑也日渐增多，诺贝尔经济学奖得主Paul Krugman称比特币的变动趋势鼓励了囤积行为。而电子货币的先驱Stefan Brands认为比特币的“金字塔”结构倾向于奖励早期参与者。

## Christopher Soghoian——互联网时代的黑客侠盗

Christopher Soghoian喜欢寻找大型机构的安全问题，通过精心安排一系列事件公开羞辱它们，促使其作出改进。



Google、Facebook、Dropbox、AT&T，还有美国联邦政府，都曾经是他的目标。

30岁的Soghoian，目前在印地安那大学伯明顿分校攻读博士学位，他几乎一整天都在电脑前度过，寻找各个机构的安全漏洞。每当他发现漏洞并找到解决方案，就会通知对方，给它们一个修复漏洞的机会，并且警告如果问题没有被修复，他会将其公之于众。

以Gmail为例。2006年夏天，在Google实习的Soghoian发现Google为了让用户觉得其服务更加快速，在Gmail中默认关闭SSL安全连接。3年之后，Soghoian联合了36名科技领域的领导人物，签署了一份公开信，督促Google默认采

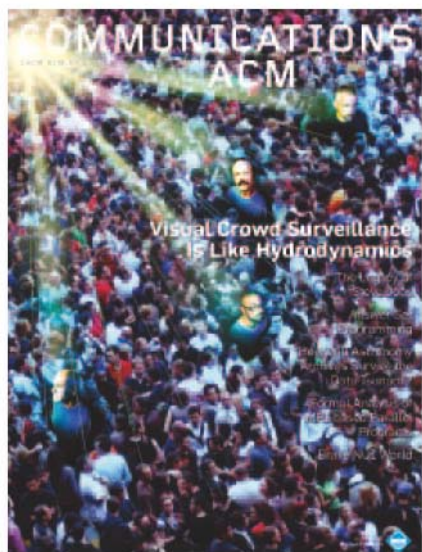
用SSL。这封公开信公布后几小时内，Google作出了正在部署SSL作为Gmail默认连接方式的声明。Soghoian还曾经搞到了Sprint Nextel一名经理的录音，其中提到这家公司向法律实施机构提供用户位置信息达800万次之多。在这段录音公开后，这件事例被联邦上诉法院第九巡回审判庭援引。

去年春天，他和几位朋友发现了Dropbox隐私条款中的一个缺陷。这家云服务提供商没有说明他们留有一个后门，可以访问用户存储在云中的数据。他在Blog上发布了这个情况，事件最终以更新用户隐私条款，声明这一后门结束。

通过这种方式，Soghoian推动这些公司

以及政府政策进行改进，并且让大众知晓其中内情。这些机构一开始会感觉受到了侵犯，但最终会意识到自己是在向好的方向迈进。很少有人能做到这一点，而Soghoian就是其中之一。

他还曾经拒绝过一家公关公司的邀约，拒绝捏造诋毁Google公司隐私政策的负面新闻，并揭露出背后的指使者是Facebook。当然，在这些活动中，他自身的名誉也遭到了一些损害，他就像是互联网时代的罗宾汉，牺牲自己为公众谋取正当的知情权。



## 自然用户界面

以微软Kinect为代表的自然界面接口开发，也许代表着鼠标的末日。因为利用微软的Kinect软件开发工具包制作出了

## Communications of ACM

2011.12

自然用户界面，独立软件开发者Kevin Connolly过去几个月里，成了YouTube上小有名气的明星。在YouTube上的短片里，Connolly演示了仅仅通过姿势移动、缩放一块屏幕上的图像，或者将它们排成三维的图像阵列。这很像是科幻电影中的场景。

微软在2011年6月做出了发布Kinect软件开发工具包的决定，为它迎来了科技新闻的曝光率，还有开发者和技术爱好者的关注。在发布伊始的6个星期里，大约有10万人下载了软件开发工具包。但是，发布细则中禁止将开发工具包用作任何商业用途，而Connolly在设计初步

的自然用户界面并试验成功后，就停止了这项工作。但是这款工具包的发布是一个信号，说明低成本的动作和景深感应技术将会在不久的将来改变人机交互的方式。以Kinect为例，它的硬件由PrimeSense公司制造，大约为200美元。许多领域的研究者都认为这个价格，相对于这样一款技术产品来说，是具有革命性意义的。

自然用户界面的下一步研究有可能是通过游戏应用来进行的，而且越来越依赖于动作感应技术。自然用户界面的发展并不意味着其他用户界面的消亡。对于作家来说，键盘可能是最适用的用户界



面；而对于船体和机翼设计这样的工作来说，三维的自然用户界面会更好。

下一代的用户界面设备必须具有三个条件：消费者能够接受的价格，简单的用户界面设计，基于该平台软件开发的方便性。PrimeSense公司推出的OpenNI驱动中，包含了一系列API，允许用户对Kinect的传感器进行操作，方便了各种应用的开发。

这种情况下，也有一些机器人研究者提出了自然用户界面的新概念。既然我们现在能够借助低成本的传感器技术让机器人感应周围的环境，那么用户界面的研究重点就应该转移到如何让机器“思考和理解”人类的意图，并和人类沟通上来。

## 匿名不等于隐私保护

我们生活在一个信息泛滥的年代。网站上的每一次点击，社交网络上的每一个联系，我们发布的每一项个人信息——线上线下的一举一动都被记录并分析，这些监视机构中，包括互联网服务提供商、广告代理，还有信用机构。随着信息量的急剧增长，这些保有信息的机构有责任保护我们的隐私，特别是在转售或交换这些信息的时候。

目前的隐私保护技术大部分都采用匿名化方法，移除一些个人特征数据（如姓名）之后，得到匿名数据集来避免侵犯隐私。这种方法在学术领域和产业实践中都很流行。但是匿名化并不能很好地实现隐私保护的目。有很多问题亟待解决，比如这些敏感的个人信在隐私泄露中占多重的分量，对方也许有能力获取匿名化之前的数据，对方也许有能力和足够的信息将数据反匿名化。很多现有的隐私技术都没能很好地回答以上

问题。

信息的急剧增长产生了很多与个人有关的敏感数据，我们需要对隐私风险有新的理解，认识新的隐私攻击方式，并且制定新的防御策略。Lars Backstrom, Cynthia Dwork和Jon Kleinberg对以上问题进行了研究，证明了数据匿名化的脆弱性，针对社交网络提出了选择匿名化节点的新方法，大大地改观了我们对个人特征信息的理解。首先，他们对匿名化的意义进行了研究，表明了简单地移除姓名等个人特征信息并不等于隐私保护。其次，他们对社交网络中的匿名化进行了细致的理论研究，指出基础的社交网络拓扑结构即可定位一个人的身份，并且在现实的大型社交网络LiveJournal中实验评估了他们的反匿名化识别技术。他们还研究了匿名化面对的一种新攻击方法，攻击方可能会有意社交网络中散布随机链接地址，用以判断网络的的拓扑图，从而用于身份识别。匿名化不能有效防御类似的攻击，因此我们需要更先进的隐私保护技术。

## 重用代码的许可证风险

重用现有的软件工件，来进行新软件开发，是节约成本和加速上市时间的好办法，而且能提高软件的质量。在软件开发中，代码是最常被重用的部分。研究者发现，商业软件开发中出现了一种新的重用方式，就是“网络代码”的重用。这里的“网络代码”是指从因特网上免费下载的代码，这些代码以部件或组件方式存在，而且没有原作者的授权。很重要的一个例子就是可以公开获得的开源软件。网络代码通常被授权可以用于商业软件开发，这对许多公司来说具有很大的吸引力。尽管如此，网络代码很

少处于公众领域，而且有许可协议，需要满足一定的条件才可重用。这些条件不一而足，有时可能会要求声明原作者对代码的归属权。还有更严厉的许可协议，如GPL要求集成GPL代码的软件也要遵循该许可协议。对于重用这些代码的公司来说，这些许可协议带来了法律和经济上的风险。

有一些公司会系统地重用网络代码，它们会权衡重用代码的收益和风险，并妥善地处理许可协议问题。而相对于系统地应用网络代码的公司，只有在特殊情况下才重用这些代码的开发者通常不会注意代码的许可协议。他们的个人行为可能会给公司造成麻烦。有研究者在全球范围内对869名职业软件开发员进行了调查。这项2009年的调查显示，59%的程序员认为在特殊情况下重用网络代码“比较重要”，而只有12%的程序员完全没有在特殊情况下重用过网络代码。有1/4的程序员根本不知晓网络代码许可协议和他们的法律责任，而软件公司也完全没有关于采用网络代码的相关政策。基于以上调查结果，软件公司必须采取措施，清查代码库中存在的网络代码，并且意识到这些代码带来的法律和经济后果。

（感谢译者卞斌支持）

# 挑战下一个大家伙

## 写在2011超级计算大会之后

文 / 詹剑锋

超级计算大会(SC)英文的全称是“International Conference for High Performance Computing, Networking, Storage and Analysis”。SC是高性能计算领域的盛会,是学术研究、工业实践与技术讨论结合的完美典范。SC每年都吸引了超过一万人以上的参会者。

2011超级计算大会于2011年11月12-18日,在美国西雅图召开。这是一个美丽而充满活力的城市,微软总部、波音总部、星巴克总部和亚马逊总部都设在这里。我们一行四人在阴雨连绵的11月见识了西雅图红叶的美丽,也感受了超级计算的魅力。这篇小文记录了我们参会的所见和所思,分享给大家。

### 异构

一句话总结本次超级计算大会的主题,那就是异构计算——GPU异军突起。在详细分析了超级计算排名表TOP500的历年数据后,SC的一篇技术论文下了个大胆的结论,未来几年如果不是异构就别玩了。

在推动GPU计算中,中国人的贡献不可忽视。中科院过程所的GPU集群、中科院计算所的曙光星云(曾经位列超级计算机第二名)和国防科大的天河一号(曾经取得超级计算机冠军),都见证了中国人在GPU上所做的通用计算的探索。有趣的是,Nvidia的销售骄傲地告诉我们,美国的BlueWater高性能计算机也将走CPU和GPU相结合的技术路线。

光有裸机,缺少应用是不行的。应用的繁荣需要编程工具的助力。SC主题演讲上Nvidia CEO

Jen-Hsun Huang透露,Nvidia联合多家企业推广“CPU+GPU”异构编程标准:类似于OpenMP和OpenACC允许并行程序员,向编译器提供简单hints,以便编译器标识哪一部分代码需要加速,而不需要用户修改底层代码。不少并行计算专家认为,未来在并行编程工具方面,“MPI+OpenMP”将于相当长时间内有支配性地位,GPU的引入可能会让OpenACC系统起着补充作用。

与此同时,一些新的力量也在蓄势待发。如Nvidia结合ARM处理器和GPU,试图以低能耗、高效率,撼动Intel在超级计算机中的地位。GPU和ARM都在推广虚拟化方案。在这其中的新产品,有可能使高性能计算的主流技术进入更为通用的数据中心领域。

### 京

超级计算机的TOP500排名也是SC上的话题之一。中国的天河一号在领先一段时间后,被日本的“京”高速计算机超越。



排名TOP500第一的日本“京”速计算机展台

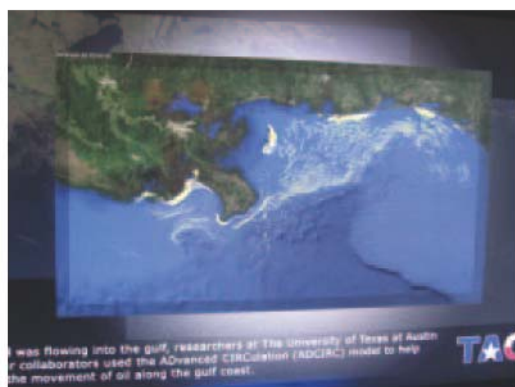


“京”（K Computer）是富士通与日本理化学研究所共同开发的超级计算机。2011年11月，“京”全部组装完成，其LINPACK性能达到10.51PFLOPS的纪录，并行效率为93.2%。

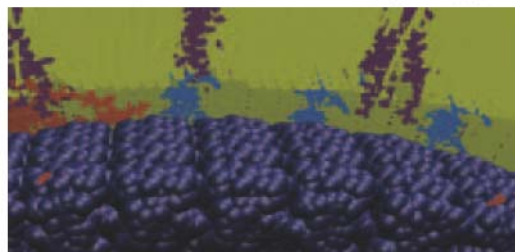
“京”系统配备了864个机柜和富士通生产的88128颗八核心SPARC64 CPU。“京”给人留下印象最深的是其网络系统，众多处理器通过名为“豆腐”（Tofu）的专用六维mesh/torus网络连接在一起。

## 差距

比较中外差距，超级计算应用是落差最大的。会上展示的可视化模型牢牢吸引了我们的眼球。美国的大学和研究所精心选择了一系列超级计算应用，如心肌、太阳系磁暴、银河系早期形成和红细胞模拟等等，极炫地展示了他们在超级计算应用方面的深度与广度。



University of Texas at Austin完成的墨西哥海湾石油污染模拟



中科院过程所和计算所完成的H1N1猪流感病毒数据细节图

在本次会议的唯一一场主题演讲上，Jen-Hsun Huang发布的消息让我们稍有欣慰。他列举了最重要的GPU计算进展：中科院过程所在数千个GPU上完成了H1N1仿真模拟。据我们了解，该工作由中科院过程所葛蔚研究员负责，中科院计算所朱登明副研究员带领课题组完成了可视化部分。

## 数据中心

尽管是高性能计算的会议，我们也看到了不少数据中心技术的展示。在一个工作坊的演讲中，微软副总裁Dennis Gannon总结了当前工业界应用的每秒钟需要处理的数据规模，令人印象深刻。这些应用也展现了数据源的多样性，从Twitter的实时数据流、粒子加速器、射电望远镜，到社交网络游戏，充分显示了数据中心已成为重要的基础设施。

联想到中科院计算所正在开展的高通量服务器研究计划，我们相信数据中心也将和高性能计算机一样成为重要的高端计算力量。遗憾的是目前缺少一个或者一组基准测试程序来评价数据中心，否则将会引发这个领域的火爆发展。

我相信未来高性能计算和数据中心领域的相互融合会加强。这种判断来自于我对云计算和高性能计算的互动分析。以云计算为例，几个核心技术都可以从高端计算中发现影子。虚拟化技术早在20世纪60年代IBM的大型机上已有成熟应用。Cluster（集群）技术本身就是高端计算普及化的产物。在云计算浪潮中，集群的应用跳出了科学计算的圈子，进一步应用于数据处理和分析。MapReduce可以看到MPI的身影。传统的资源管理在云计算系统里得到普及。

高性能计算机和数据中心结合到一起，将会是我们在云计算的未来里遇到的又一个大家伙，既是机遇，又有挑战。

## 期待

2012年我们会再来SC，国内几个参会单位的人员也会更积极参与。如果说期待，有以下两点：一、希望有更多的中国参展单位。在学术报告方面，中科院计算所、网络中心都有大会报告论文，但与日本和韩国的同行相比，中国的参展单位偏少。二、尽管有一些数据中心技术的展示，但还未形成主流，期待数据中心领域有更多的展示，能与其基础设施的地位相匹配。P

（作者唐剑锋为中科院计算机所博士生导师）



# 性能追求，永无止境

## Velocity China 2011观感

记者 / 佳琪

2011年12月7-8日，北京，O'Reilly和淘宝网联合举办了第二届Velocity China会议。与去年相比，此次会议议题特点鲜明，主要体现在以下三点。(1) 高性能移动互联备受关注；(2) 结合自身业务的定制化分享增多；(3) 流行度高的新工具新技术介绍多，且都有实际案例。参加此次会议，我们可以清晰地把握近段时间Web性能和运维方面关注点的变迁。

### 移动互联

移动互联网飞速发展，用户对移动设备上网性能也提出了更高的要求。然而，现状是不令人满意的。据调查，85%的用户希望在移动设备上的下载速度至少能和家中计算机一样快，如果网站在移动设备上的表现太差，有近半数的用户认为他们不会再访问了，三成用户则因速度体验太差，转投竞争对手的网站。

反过来，如果能提高移动设备上的用户体验，可以带来极大的访问提升，来自Strangeloop Networks的Joshua Bixby就利用大量来自一线的案例分享，给出最新的数据，展示了现实世界的公司如何优化移动网站性能，以及由此带来的商业KPI戏剧性的提升。

那么，面对用户的高期望值，在移动终端设备迅速占领市场的今天，基于客户端的Web应用如何做到更好呢？

Steve Souders就在演讲中，详细介绍了优化移动端用户体验的技术方法。在Yahoo! 和Google的从业经历以及Yslow和Firebug的开发经验，让Steve

Souders格外重视移动互联网的优化。他指出，在移动互联时代，Yahoo! 曾经的14条铁律已经不再完全适用，更多的是依靠App自身设计。【注：关于Web高性能，Souders还写过两本书，*Even Faster Web Sites*和*High Performance Web Sites*，值得关注高性能的开发者研读】。

同样，来自Facebook的魏小亮总结了在移动互联网的性能优化方面的经验和教训。针对新一代Facebook移动平台，他分享了三个方面的内容：新平台的接口和功能、平台上移动应用的研发流程、平台为移动应用提供的传播支持。

### 一线经验

公司的业务不尽相同，即使同样的技术，每个公司根据自身特点制定出来的结果也大相径庭，如何取长补短？在会议中，大家得到了很好的交流。

来自去哪儿网的林浩和刘连春，在《机票实时搜索引擎的优化》的议题中，介绍了他们如何调整架构、优化流程、加快数据解析速度、减少内存占用和降低带宽消耗的过程，以及在其中积累的大量性能优化的实战经验。

近年来，淘宝网的业务呈指数级增长，每天数十亿的PV对前端Web服务器的性能提出了非常严苛的要求。淘宝工程师朱照远和王晓哲因此介绍了如何做到Web服务器的高性能，同时保持其安全性与可运维性，分享淘宝网是怎样通过定制开发Nginx服务器内核以及开发有效的模块来达到这点的。

同样的情况，对淘宝的基础架构也提出了挑战。

这促使企业必须寻求“稳定、高效、低碳”的基础架构方案，淘宝的章文嵩博士对此介绍了非常具有实战价值的内容。

豆瓣网的用户已经达到了五千万，有大量数据需要分析计算。MapReduce类框架在大规模数据计算上有很好的伸缩性，但在实现需要迭代类算法时效率比较低。UC Berkeley的AMP实验室设计并实现了Spark来解决迭代计算问题，建立在通用的计算资源调度框架 Mesos之上。Dpark是Spark的Python克隆，实现了Spark的绝大部分特性，使得可以直接用Python进行MapReduce或者迭代式计算，这正与豆瓣偏好使用Python的技术口味相吻合。豆瓣工程师刘洪清介绍了豆瓣网正在积极尝试用Dpark来取代Hadoop和部分MPI来实现日志分析和各种数值计算。

## 新工具/技术

往往一个性能的瓶颈会因为新技术的应用迎刃而解，所以关注新工具新技术是每一位工程师必须做的事情。

Node.js是最近关注度比较高的服务端开发框架，具有并发性能高、开发简便的特点。淘宝网的工程师们是Node.js的鼓吹手。在演讲中淘宝工程师廖恺结合Google App Engine这种比较流行的应用托管方式，介绍了CnodeJS社区最近将这两种技术结合，发布的Node App Engine项目。在主题中，他主要介绍了Node.js的特点与发展，以及Node App Engine的功能特点与设计思路变迁。

Yahoo的Fabian Frank介绍的也是Node.js。不同点是，在真实案例分析中，他对比了Node.js与PHP服务端性能。并且做了实用、可操作的Node.js性能和可伸缩性分析，用户可以据此来评估应用是否可以采用Node.js运行。此外他还展示了Yahoo的Node.js MVC框架——Mojito（在2011年12期的新产品新工具栏目中有过介绍）。此框架在Yahoo内部运行良好，并在计划2012年12月开源。

从Web1.0时代至今，大家一直在尝试各种方法，提高用户对互联网产品的浏览速度，在Web应用多样化、复杂化的今天，新浪微博团队向



来自世界各地的Web运维专家分享自己的经验

Facebook等国外先进网站学习、借鉴，试图通过对现有资源应用方式的改变，来达到提升用户浏览速度的目的。使用BigPipe提升浏览速度就是他们的成果之一，来自新浪的吴侃介绍了流水线技术在新浪微博的实践。

SeaJS是专注于浏览器端的模块加载器，它的拥护者们致力于打造基于此项技术的JavaScript生态圈。使用SeaJS可以获得愉悦的调试体验和便捷的打包部署工具。此技术在腾讯、百度、盛大和网易等网站也都有应用。来自淘宝的王保平就介绍了基于SeaJS的高性能网站开发和优化实践，并用实例演示了SeaJS在“淘宝商品详情页”、“一淘直通车”和“聚划算”等重要系统中的应用。

## 结语

在此次大会上，我们看到Web技术层出不穷，让人眼花缭乱，但广大程序员万万不要看到哪些技术成为热门话题就盲目去学习、效仿。在开始一项新技术之前，我们要深思熟虑一下，这门技术在哪里可以应用得到？是否有很迫切的需要？把精力投向于那些有生命力的技术和自己工作相关的技术，并与同行多交流和分享经验，可以让程序员职业生涯走得更有意义。P





# 黑莓突围

策划 / 孟迎霞

2011年12月7-8日，黑莓亚洲开发者大会在新加坡新达城国际会议中心召开。此次会议，黑莓首次正式展出新一代移动平台BlackBerry 10、BlackBerry PlayBook OS 2.0开发者测试版及更多升级版开发工具，力图在这场移动大战中突围成功。本刊特邀与会开发者，众议黑莓新移动平台及其背后的技术策略。

## BlackBerry 10下一代移动平台

文 / 耿聿

12月7-8日，黑莓亚洲开发者大会在新加坡召开，RIM再次展示了其强大的技术实力，并重点展示了BlackBerry 10（2011年10月的BlackBerry DevCon America上被称为BBX，后因商标侵权问题而改名）。它采用开放策略，并整合多种领先于现有主流平台的新技术，是名副其实的下一代移动平台。下面介绍该系统中值得关注的地方。

### 以WebWorks为开发环境，QNX为系统内核

首先要谈的是其采用HTML5+JavaScript的WebWorks开发环境。显然，RIM打算将其打造为BlackBerry 10上的首选开发环境。虽然HTML5被寄予厚望多年，被视为移动开发的未来，可现在主要还是基于平台原生开发环境的应用程序。开发者只能利用第三方平台如PhoneGap进行开发，因此在性能、平台支持度、安全限制等方面都有

很多局限。而WebWorks由系统厂商直接提供，并得到了重点支持，所以开发者不需要有任何平台支持方面的顾虑。一方面，RIM提供了一整套功能强大的开发工具，包括基于命令行的SDK、Eclipse和Visual Studio开发插件以及Ripple模拟器等。另一方面，WebWorks提供了丰富的API，JavaScript脚本可以方便地访问各种系统功能，其中包括BBM、Map等服务，还包括Camera、Acceleration等硬件资源。此外，WebWorks还提供强大的WebGL绘图接口，这让直接使用其来开发2D/3D游戏成为可能。

第二点要谈的是新系统的内核。BlackBerry 10的核心是QNX操作系统，它是第二代微内核操作系统的代表，从架构上讲比现有主流操作系统更先进，具有更好的稳定性和可移植性。其本身也是一个成功的RTOS，广泛应用于各种嵌入式设备。因此开发者不用过分担心新平台的稳定性问题。进一

步,可以大胆想象,我们的程序将快捷地移植到其他基于QNX的嵌入式设备上,各种QNX设备可以方便高效地进行连接与合作,比如平板与汽车系统的合作、手机与家电的互联等。这里还想提到的是,QNX是一个Unix-like的操作系统,因此其NDK对UNIX/Linux上的C/C++程序员来说很容易上手。同时,很多现有C/C++程序库的移植也相对比较容易。RIM正在进行Qt的移植工作,而其原生的Cascades UI Framework也是基于Qt,这必将成为新平台吸引来大量原Qt开发者。

## 发展两大策略：开放和云服务

除了Qt, OpenAL、Cocos2D、Lua等大量优秀的开源项目也正在或已被移植到BlackBerry 10平台上。开放是RIM发展BlackBerry 10的重要策略,也是新系统最有吸引力的地方之一。RIM与苹果比较像,从软件到硬件再到服务都是一家公司完成。这样的好处是具有良好的集成能力,坏处则会让开发者对其封闭性感到担心。而RIM正打算通过逐步开放的方式来消除这一问题。现在WebWorks SDK已经开源,同时,RIM还建立了多个开源项目,如WebWorks-Community-APIs、Alices.js等,以吸引广大开发者为其平台提供优质源码。

BlackBerry 10上的另一主要策略是大力发展云服务,RIM有计划将各种服务通过其社交平台BBM有机地结合在一起,形成一个真正的云服务平台。同时,RIM还计划提供统一的云存储服务。以上服务的接口都将整合进其开发环境中,开发者



可以方便地基于这些服务开发自己的应用。

虽然有先进技术和正确策略,可能否最终成功,还要看黑莓执行力。对此我有些担忧,因为已经多次出现关键技术发布被推迟的历史。比如Cascades UI最早宣称2011年秋天发布,可至今仍没给出准确发布时间。BlackBerry 10中两大开发环境WebWorks和NDK仍没得到有机结合。如果想为WebWorks提供扩展功能,现在只能选择Adobe ActionScript编写,因此WebWorks仍然不能利用很多NDK的强大功能。据我了解,RIM已有计划支持这项功能,但现在还不能给出任何时间表。此外,具有核心地位的BBM服务仍然未集成到PlayBook上,因此开发者仍无法为其平板电脑开发BBM相关应用。

总之,BlackBerry 10对于开发者来讲非常有吸引力,我真诚希望RIM最终可以以BlackBerry 10为核心打造出一个优秀的移动开发生态系统。P



耿聿

资深移动平台开发工程师。2008年加入Nokia,从事Symbian操作系统的开发。2011年加入VMware,负责在iOS、Android、BlackBerry等移动平台上VMware虚拟桌面产品的研发工作。

## 黑莓 & 企业级开发

文 / 艾晨

本次大会上,RIM终端应用平台及工具副总裁Christopher Smith介绍了BlackBerry 10这个新平台。它基于QNX操作系统,与PlayBook使用相同的系统,使手机和平板可以保持更好的一致性。下面我想从企业级开发角度谈谈我的看法。

### 提供BES解决方案 移动开发更快速

随着移动信息化的发展,当前企业用户关注更

多的是,能否把现有应用搬到手机或平板上、使用是否方便、开发是否迅速、应用是否安全等因素。依靠黑莓BES解决方案,企业开发者在开发过程中,无需额外考虑移动设备安全性,数据传输的及时性、可靠性、稳定性,程序的部署升级,以及不同的网络连接方式,因为BES已经很好地解决了这些需求;黑莓开发者只需要考虑企业应用的内在逻辑、界面以及其他使用细节问题,将更多精力花在企业应用上。



## WebWorks 降低开发难度

使用黑莓开发平台WebWorks进行开发,是一个很好的选择。WebWorks支持HTML5,本次大会的一些分会场展示了一些非常炫丽的效果,均是用HTML5开发。以前用Java开发黑莓手机程序时,要花很多时间在界面布局、页面跳转这些方面,随着黑莓手机操作系统的不断升级,以及对WebWorks更多的支持,这些将更加容易。

我们还注意到,越来越多的黑莓手机支持NFC。演示中RIM使用带有NFC模块的黑莓9900打开通讯录,并进入某联系人的详细信息,这时与另一部带NFC模块的黑莓9790的背面互碰一下,好像握手一样,叮咚一声,这个联系人的信息就传到了9790上,非常方便。其实NFC或RFID技术,在企业中已被广泛使用,比如在物流业、医疗业中被

用作电子标签。

另外,统一通信和设备管理,也是RIM提供的重要功能。其统一通信解决方案MVS(Mobile Voice System,移动语音系统),即将引进中国。在设备管理方面,RIM最近公布了一款新产品BlackBerry Mobile Fusion,它将为iOS、Android等平台提供高级企业移动管理方案。

黑莓在企业级市场一直占有领先地位,不仅仅是因为黑莓手机本身的商务特质,更多缘于黑莓解决方案提供给企业开发者的整套平台。P



艾晨

神州数码黑莓技术支持,黑莓培训师,致力于黑莓企业级应用开发。

# 黑莓:值得重视的移动平台

文 / 林金理

很多开发者都会问,如何选择进入一个平台?作为跨平台移动应用发布商,摩摩网的衡量标准是:目前能否获得盈利和机会,以及未来的发展潜力。我认为这是最直接、最简单的评估点,不用反复考虑,把更多剩余时间用来研发、营销、试营运、反馈和调整。

在刚闭幕的黑莓亚洲开发者大会上,黑莓清晰勾勒了其线路图。OS7仍然是目前和未来短期内非常重要的平台,也是开发者可以获得盈利的较好平台。在下一代BlackBerry 10系统推出之前,开发者会有充足的机会来准备和升级。

我们摩摩网从2007年开始进行黑莓平台的内容开发——手机主题,之后又开始做应用内容的研发,截至2011年11月,我们在黑莓官方应用市场BlackBerry App World上的下载量累计超过4000万次,每天新增下载4万次;在这样的产品矩阵下,我们在黑莓的业务一直保持盈利和成长。

黑莓是摩摩网非常重视的移动平台,本次大会

上,我们发布了两款针对黑莓平台的重量级首创应用:MMMOOO Gif和桌面宠物精灵,前者在黑莓上属首创和唯一的Gif生成软件;后者是利用我们的3D产品创作能力,结合独特的黑莓接口,使用炫技术研发出的产品。

总之,我的观点是:黑莓的表现比外界想象的要好很多,并且它也在进行积极升级和突破。

移动互联网时代,各个平台厂商都在争霸天下,而应用开发和服务提供才刚刚开始,目前绝非最后的战场。作为开发者,要把短期和中长期结合好,把开发产品与团队建造好,把企业营运平衡好。一切还在拭目以待! P



林金理

摩摩网创始人、CEO,毕业于西南财经大学。大学期间创建的校园电子商务站点流行于成都各大高校,曾于2003年代表所在大学参加“挑战杯”。

# 2011 Qt开发者大会： 凝聚开发者的利器 描绘生态系统新蓝图

记者 / 赵红 付江



12月12-13日，由CSDN和诺基亚联合举办的2011中国Qt开发者大会在北京成功举办。本次大会吸引了超过千人的应用开发精英到现场，与专家共同探讨Qt应用开发之道。大会内容包含了专场技术培训、主题研讨会、案例研究和合作伙伴服务，以满足各类开发人员的特定需求。作为凝聚开发人员的最佳利器，将持续投入把Qt打造成应用开发的核心组件，助力“下一个十亿用户”战略，确保投资Qt的开发者在未来能与诺基亚发展共存。

第一天下午的专场技术培训内容包括Qt Quick、图形元素与动画、带数据模型的列表和网格以及其他元素概述、RssNewReader——实时应用程序的创建等热门领域。

第二天上午的主会场演讲重点介绍了Qt战略定位、开放管理特性、Qt5路线图及新版本规划和各种实战案例。而下午的分论坛内容丰富多彩，涉及MeeGo与Symbian中的Qt跨平台开发、Qt Mobility、Qt Quick深度解析、在Qt应用中加入多媒体、WebKit与HTML5及Qt文本渲染技术。

所以本次大会从全方位技术指导到用户体验、再到推广营销等多个方面和与会者进行了深度沟通和互动交流，描绘出一个庞大完整、平衡发展的开发者生态系统蓝图。

所以本次大会从全方位技术指导到用户体验、再到推广营销等多个方面和与会者进行了深度沟通和互动交流，描绘出一个庞大完整、平衡发展的开发者生态系统蓝图。

所以本次大会从全方位技术指导到用户体验、再到推广营销等多个方面和与会者进行了深度沟通和互动交流，描绘出一个庞大完整、平衡发展的开发者生态系统蓝图。

诺基亚Qt生态系统全球总监Daniel Kihlberg

## 微观点

从2008-2011年，Qt下载量从50万增加到200万。从4.8版本开始，Qt已经支持Windows、Mac和Linux11等跨平台开发，在Qt中加入WebKit2.2、HTML5以及文本渲染等技术。未来将推出的Qt5会加强性能和先进的画面效果。

——诺基亚Qt生态系统总监 Daniel Kihlberg

Qt5将具备更多新特性，包括支持更多种编程语言、改进图像的缓存机制、空间导航，以及在触摸屏和移动装置的方面增加新的性能改进工具等。此外，Qt模块也将在特定环境下增加额外价值，包括多媒体、地图、图形化效果等。Qt5还支持包括Android、Windows、iOS、Linux11在内的多个平台。

——诺基亚Qt开发框架首席设计师 Lars Knoll

@点点点软件开发工作室：使用Qt Creator可以很容易开发基于MeeGo的N9和Symbian手机的应用。只需要简单地在配置文件中增加相应内容即可，非常方便。

@掌间无限：诺基亚Qt生态系统全球总监Daniel Kihlberg接受采访时表示，诺基亚在2011年投入了大量资源开发和推广Qt。Qt的市场流行程度越来越高，包括Mac版QQ、谷歌地球、（某些）车载应用等均采用了Qt的相关技术。

@刘维55：Qt是诺基亚开发的一个跨平台的C++图形用户界面应用程序框架，它提供给应用程序开发者建立艺术级的图形用户界面所需的所用功能。Qt是完全面向对象的，很容易扩展，并且允许真正的组件编程。



# 软硬件协同设计：云计算的另一种趋势

记者 / 董世晓

根据维基百科的描述，云计算是“通过网络将庞大的计算处理程序自动分拆成无数个较小的子程序，再由多部服务器所组成的庞大系统搜索、计算分析之后将处理结果回传给用户。通过这项技术，远程的服务供应商可以在数秒之内处理数以千万计甚至亿计的信息，提供和‘超级电脑’同样强大性能的网络服务。”

因此，云计算本质上是与超大规模、超大数据量、超大带宽密切相关的，它将海量数据的集中计算任务交给云端进行处理。从某种程度上说，云计算就像百度首席架构师林仕鼎在2011中国计算机大会上的观点那样：“比起云计算，数据中心计算在技术上可能是更合适的表述。以前的系统设计考虑的主要是单机环境，由单机到数据中心，很多设计原则发生了根本的变化。”

## 云计算的两种趋势

我们已经清晰地看出，经过这几年的发展，云计算渐渐显现出了两种趋势。

一种趋势是以Google为代表的公司，通过使用MapReduce架构等软件方面的技术手段，将数据拆成小块，分配给大量的廉价PC服务器计算后再重组回来。这种方式的特点是先期投入比较低，但存在PC服务器故障频繁、数据中心占地面积过大、电费开支过高等诸多问题。

此外，我们不能寄希望于每家公司都像Google、Facebook、阿里、百度那样，拥有强大的技术力量，自行搭建云计算平台。对一般企业而言，它们更多地是希望能有一种省心、省力的方式，解决自己在大数据时代遇到的发展瓶颈。

正因如此，云计算的另一种趋势——软硬件协同设计也浮出水面，并逐渐得到了认可，其中的代表便是IBM的POWER虚拟机。POWER虚拟机搭载了IBM自己的Unix操作系统——AIX，能够最大程度地发挥硬件性能，使软硬件的协作达到极致。

照理说，软硬件协同设计应该是业内特别是硬件制造商早有的共识，但时至今日为何只有IBM在推进呢？IBM系统与科技部大中华区Power Systems服务器产品部总经理韩忠恒给出了答案：“软硬件协同设计需要投入大量的资金和人力。而IBM之所以能够一直在POWER/AIX研发方面持续投入，是因为非常看

好这种技术，并且将云计算看成一种绿色技术。因此，每年会投入60亿美元，进行POWER/AIX方面的研究。”

## 也说POWER/AIX的历史

POWER是IBM服务器、工作站和超级计算机的主要处理器，起源于801 CPU，是第二代RISC处理器。它与AIX结合在一起，打造出了第一台RISC 6000系统，也就是RS/6000系统。至今POWER已经有了七代产品，目前最高端的是POWER7。

尽管公众对POWER/AIX比较陌生，但对1997年的那场人机智力大战则再熟悉不过了：国际象棋大师卡斯帕罗夫以25:35不敌一台名叫“深蓝”（Deep Blue）的RS/6000SP型超级计算机（基于POWER2架构）。“深蓝”装有32个并行处理器，每秒能分析2亿步棋，同时存储了几乎世界上所有的棋谱，对于卡斯帕罗夫过去下的每一局棋都了如指掌。“深蓝”的程序设计人员利用特定的评价函数和AIX系统以及POWER强大的大规模并行处理信息能力，使这场对弈完全演变成了一种计算。

那次大战之后，POWER似乎总是在公众的视线中若隐若现、不温不火。

终于，2011年2月14日，POWER/AIX再一次席卷了公众的视线。采用IBM Power 750服务器的“沃森”（Watson）计算系统亮相全美最受欢迎的智力竞赛节目《危险边缘》（Jeopardy!），并与该节目史上最强大的两位答题高手连续三天同场竞技，最终以绝对优势赢得了人类历史上第一次人机智力问答比赛的冠军！

对此，韩忠恒表示：“从‘深蓝’到‘沃森’，都以Power服务器作为支撑平台——‘深蓝’的成功得益于POWER2大规模的并行计算能力，而‘沃森’则充分体现了POWER7海量计算与实时分析工作负载优化系统的特性。IBM服务器再次说明：计算机在帮助人类更好地生活与工作、建设‘智慧的地球’上，所蕴藏的无限潜力和不可估量的重要意义。”

据悉，“沃森”在3秒内处理海量并发任务和数据的同时实时分析信息——检索大约2亿页的内容（约一百万册书籍的价值），分析数以百万计的信息碎片，并且根据它寻找到的证据提供最佳答案。很显然，这些靠的是不只是硬件能力，AIX也功不可没。



IBM系统与科技部大中华区Power Systems服务器产品部总经理韩忠恒对POWER7信心百倍

似乎这样的展示还不够直观，于是2011年12月15日在北京798艺术区内，上演了一场POWER虚拟机与x86物理机的应用PK大赛。现场共进行了两场测试：汉铭通信手机报应用和创业软件区域医疗应用，前者是AIX对决Linux，后者是AIX对决Windows。从结果来看，POWER优势明显。

## POWER/AIX在中国

既然POWER/AIX如此强大，自然就打动了国内外众多客户，别的不说，POWER现在已然悄悄吞噬了中国60%的服务器市场。下面仅举两个代表案例。

### 统计云

2010年初，国家统计局数据管理中心开始将其核心业务系统建立在以POWER为基础架构的云计算平台上，以确保第六次人口普查和企业直报两大应用系统顺利运行，同时通过横向和纵向的扩充来支撑、整合更多的应用。一期项目建设，通过采用6台Power 595小型机组成的物理资源池，其中4台配置40颗CPU的Power 595小型机实现数据库的集群，分别运行两个数据库实例（人口普查、企业联网直报），形成数据库资源池；其余2台配置64颗CPU的Power 595则实现中间件虚拟化，实现应用

服务器资源池。同时，借助PowerVM高级虚拟化技术、Tivoli Monitor资源监控软件以及Tivoli Provisioning Manager资源供给管理软件，实现了对核心业务系统监管及计算资源自动化调配的资源运行平台，成为国内领先的基础架构云平台。

这个平台通过多项重大统计任务的实践检验后证明是成功的，13亿多人口的复杂表格，两分钟内就可以统计完成，而过去则需要一天甚至更多时间。

### 电信IDC云

贵州移动云计算平台始建于2009年底，3个月后便开始运行各类客户应用。该平台基于POWER系统，通过PowerVM服务器虚拟化技术与SVC存储虚拟化技术，将物理资源Power服务器以及XIV网格存储进行全面虚拟化，形成虚拟的计算资源池和存储资源池，再通过定制开发统一管理平台将TPM、TSM、ITM管理模块进行整合，最终形成统一的资源管理服务平台。

这个平台不但解决了传统的“烟囱式”IT建设模式，而且将基础架构和IT资源作为服务交付，实现了全面的、真正的虚拟化IaaS平台，有效地实现了异构资源的整合，形成按需分配的弹性资源使用模式，提高资源利用率，大幅降低IT资源的建设投入及运维成本。P





年度十大热词：平台 | 移动 | 创业 | 大数据 | 编程语言 | 乔布斯 | 用户体验 | 并购 | 创新 | 泡沫

# 平台

文 / 刘江

领域热词：生态系统、数据、信息运营商、云计算

2011年最热的关键词恐怕非平台莫属。原因很简单，一年来发生的重大事件几乎都与平台息息相关。

在芯片层，ARM架构已经占据移动设备的大部分份额，对曾经无可争议的霸主Intel造成极大压力。其中Nvidia的成功转型尤为典型，它借助基于ARM的Tegra打了场漂亮的翻身仗，2011年11月发布4核移动芯片，巩固了自己与高通、TI、三星等一道作为一线ARM厂商的地位。而Tilera等众核芯片、GPGPU等异构计算架构在云计算环境下也具有很大潜力。

在操作系统层，Windows 8和Mac OS X Lion两个新版本的用户界面已经不约而同地受到移动设备的强烈影响。而iOS和Android两大移动平台迅速崛起，不仅大幅抢占了Windows Mobile/Windows Phone、MeeGo、webOS、Symbian、BlackBerry的市场份额和生存空间，不动声色地将PC降低为普通设备，更使Google和Apple两大平台公司如虎添翼，HTC、三星、华为、中兴一千新锐野蛮成长，同时却让微软、Intel、惠普、诺基亚、摩托罗拉、索尼、黑莓等巨头都纷纷进入多事之秋，换帅的换帅，收购的收购，裁员的裁员。

## 什么是平台？

我们最熟悉的是**技术平台**（或称开发平台、计算平台），指软件应用能够在其上运行的软件和硬件环境，硬件体系结构、操作系统、API、运行库、编程框架等都是常见的平台。随着Web的普及，浏览器及其插件如Flash和Silverlight也都

成为平台。企业软件开发中，又有WebSphere、JBoss等中间件和各种面向领域的业务平台，也都是平台。

Web 2.0、云计算发展起来后，为了共享数据，提供更无缝的体验和增值，很多互联网网站和服务开始提供开放API，第三方开发者可以在其上开发软件应用，访问平台的数据或其他功能。这就是我们已经听得耳朵起茧子的开放平台。

这些当然都是重要的平台。但是，从整个行业来看，平台还有一层更重要的含义。

2011年2月，诺基亚新任CEO Stephen Elop一份语气非常罕见的内部备忘录曝光。Engadget评论这是有史以来最激动人心和有趣的CEO备忘录之一。其中他向芬兰的同事们宣布，已经决定与微软战略结盟，并同时将Symbian、MeeGo和Qt三大拥趸颇众的平台打成遗留系统。还记得他是怎么说的吗？

“设备之间的竞争已经演变为生态系统之间的竞争。我们的竞争对手并非通过设备抢占市场份额，而是通过一个完整的生态系统。这意味着，我们将不得不决定建立、促成或加入一个生态系统。……

“诺基亚，我们的平台着火了。”

虽然外界对Elop一直争议不断，而且将近一年之后的今天，他还没有交出一份满意的答卷，但他却说出一个至关重要的观点：**平台就是生态系统**。他还这样明确地给出了生态系统的定义：

“不仅包括设备的软硬件，还包括开发人员、应用、电子商务、广告、搜索、社交应用、地理位置



服务、统一通信及其他许多内容。”

Elop很可能是看到了未来竞争的残酷性和诺基亚的战斗能力不足，才最终选择了结盟，加入一个现成的生态系统。

## 平台之战

全球范围内，有实力成为全局性平台级生态系统的公司，其实并不多。2011年5月Google董事长Eric Schmidt在AllThingsD D9会议上提出了“四驾马车论”，称未来十年驱动创新的主要力量将是Google、Apple、Amazon和Facebook四家公司，它们将史无前例地在全球范围内展开竞争。除了这四大之外，他还提到PayPal和Twitter作为候补，却明确将微软排除在外。

这种说法虽然是一家之言，但得到了广泛认可。2011年11月《Wired》的采访中，Amazon CEO Jeff Bezos基本同意这一说法，只是认为微软应该在列。他随后还说了一段耐人寻味的话：“一个公司不应该沉迷于暂时的辉煌（shiny），因为不可持续。真正重要的是你的服务对客户有价值。一些公司还没有经历困难时期，没有得到真正的考验。”乍看shiny一词还以为他在说Apple，但最后一句又似乎是在说Facebook和Google。

《Fast Company》杂志2011年11月刊登了“The Great Tech War Of 2012”长文，讲述的也正是这四家之间的大战。

四驾马车的确是全球性平台级生态系统的最有力竞争者。它们都是消费互联网公司的代表，可以作为Marc Andreessen所说的互联网正在吞噬世界的绝佳注脚。虽然各家目前的主营业务并不特别重合，但是大势所趋，竞争所迫，这些千亿美元级别的公司都会不断扩充业务，延揽盟军，成为全能战士，变得越来越像。2011年，这一趋势已经非常明显：Amazon发布Kindle Fire直接成为iPad的劲敌，Google除了在社会化产品上与Facebook针锋相对之外，又收购摩托罗拉，加强与Apple的对抗能力。图书、杂志、音乐和影视、游戏、软件等内容的销售分成是兵家必争之地，四家都已经加入战团，虎视眈眈。在统一通信方

面，则除了Amazon之外，其他三家都已经重兵布局。而且更重要的是，它们都不会设定自己的经营边界，而且对远景的看法有诸多相似：都相信自己能够大大改善而非仅仅取代现有的各种产品和服务。

2012年，平台之战已经一触即发。

我在2011年4月撰写的“云计算：史无前例的大变革才刚刚开始”一文中写到：

“云计算将极大地改变原有的信息产业格局。计算机软硬件、通信、互联网、广播影视、新闻出版等行业中的大型平台性厂商将最早受到影响，它们之间将发生剧烈的竞争和整合。……

“由于互联网天然具有无边界服务的特性，这种竞争是全球性的，如果没有主权、国家体制等限制，理论上全世界只需要几家大型的云计算运营商和几家大型的设备提供商。这意味着，现有的几十家市值数百亿元以上美元的技术公司，最后只会剩下几家。”

这里面有一个关键词：云计算运营商，也可以称之为信息运营商或计算运营商。与同义词平台或者生态系统不同的地方是，这个说法主要针对最终用户而言。

顾名思义，所谓信息运营商，就是为付费的订阅用户提供信息服务的公司，它们与用户的关系与今天的电信运营商提供电信服务类似，而不同的是，它们经营的是全面整合的信息获取、推荐和处理等服务。简单说，下一步你可以在任何地方，通过任何屏幕、任何设备，获取你所需要的信息，并进行相应处理，系统是非常智能化的，可以做各种推荐（广告变成了有用的信息），就像现实生活中对你知根知底的小秘一样。而做到这一点，最重要的是要有**统一的ID登录系统和足够多具有黏性的用户**。用户只需要登录，系统就可以根据以往用户的行为历史，进行智能推荐（现实生活的商品、媒体内容……），从而实现无缝的用户体验。正如华为高级副总裁徐直军所说的：“计算、技术本身将最终被淡化、忽略掉，取而代之的是消费和体验的过程。”

平台之争中开发者的地位至关重要。如果你能在

收入上、用户获取和保留上、技术支持上、网络效应推广上真正地帮助开发者，建立生机勃勃的生态系统也就成功了一半。

最终，平台/生态系统/信息运营商之间的竞争，将是数据的竞争。“数据是云计算时代的石油”，谁能拥有更多数据并能做各种深入分析、善于运用，支持平台本身发展的决策，谁就将在竞争中立于不败之地。

## 竞争分析

在我看来，Google、Apple、Amazon和Facebook四家中，都有各自的软肋，这场大战的输赢还很难预料。

Google的主营业务——搜索是四家中唯一一种无需用户登录就能使用的服务，对用户的长期黏性最弱。从这个角度来看，Gmail是Google的战略资产，也难怪Google推Wave、Buzz和Google+都需要从这里入手。但Gmail的用户数大约是2.6亿，还不如3.5亿的Hotmail和3.1亿的Yahoo Mail，而且邮箱用户与其他三家的ID往往绑着信用卡或者支付手段相比，转换性也差得多。这正是Google+只能成功不许失败的原因。当然Android、YouTube和Chrome都是有数以亿计用户的终端，可以打的牌还很多。但Google现在内部的研发文化味道已经不对，反而是最大隐患。

Apple的问题在两方面，一是它本身还不是典型的互联网公司，云端的技术储备不够，这一点从乔布斯演讲时有些外行地宣扬自己的数据中心里设备都很贵可见一斑。二是主要收入来自硬件销售，内容销售占比太低，不符合长期趋势。在信息运营商的背景下，硬件是倾向于补贴和免费的，这和今天电信运营商会补贴终端类似。Kindle Fire就是很好的例子。Amazon每卖一部要倒贴20美元，但他们肯定有把握从媒体内容的销售中回收。一旦其他运营商将硬件免费赠送，而Apple又无对应之策（他们应该有），将有可能面临毁灭性打击。

Facebook目前最弱小，但路线最清晰，只要保持最大人群和目前的用户黏性，向各处发展似乎都

不会太难。唯一但是致命的例外是它对移动终端毫无控制，丧失了重要的天王山。毕竟，你在iOS设备上没法继续作为平台与应用分成的。Zynga一家独大也是隐患。在增加收入的同时，还要保证用户体验，难度也不小。

至于Amazon，最大的挑战是填补或者加强目前缺乏的各种必需服务（统一通信、社交、地图、开发者社区），并从纯粹的电子商务网站（大部分用户过来基本上满脑子想的是买东西）扩展为更通用的平台。以Bezos的长线思维，Amazon的想象空间还很大。

而微软呢，其实手里有大量的好牌——Windows用户数以十亿计，Live Messenger、Hotmail、Skype、各版本IE也都有几亿的用户。他们所缺的，是全面系统的战略和Google+那样的中枢应用，将零散在各处的用户和体验统一起来。Windows 8看来有这个趋势，但新版本能渗透到什么程度，要看产品的执行。

## 技术的影响

HTML5对各生态系统之间的竞争局势影响巨大。如果HTML5成为主流，Apple将丧失对生态系统的大部分控制，App Store的辉煌也将成为历史。但移动互联网恐怕不容易出现PC互联网上IE一统天下之后，再在这一平台上出现Yahoo、Google、Facebook等众多新贵的景象。因为那时候用户已经被瓜分殆尽了。

Siri为代表的自然用户界面+开放平台也是一种具有潜在爆炸力的技术。正如Robert Scoble在近两年前Siri被Apple收购时撰文所指出的，Siri代表了继1990年代“通过域名找页面”、2000年代“通过页面交互聚人”之后的第三代互联网交互模式：“消除网页，将API与人直接联系在一起”。如果能在技术上比较好地实现推荐引擎和前端自然语言理解，Siri真的可以成为数字秘书的具体化，从而将Web变成遗留技术。

2012年，敬请观赏平台之战。📍



年度十大热词：平台 | 移动 | 创业 | 大数据 | 编程语言 | 乔布斯 | 用户体验 | 并购 | 创新 | 泡沫

# 移动

文 / 陈博

领域热词：创业、盈利、平台、应用、黄金时代

2011年，是移动互联网的井喷之年，移动互联网正展现出其对未来的统治力。从这点看来，开发者们投身移动创业似乎是大势所趋。的确，在这样一种创业成本达到新低，谁都有机会一举成名，复制《愤怒的小鸟》的神话？谁不想奋力一搏，在这个“黄金时代”探求自己的宝藏？

但任何时代都逃不开“盈者通吃”的丛林法则，市场上的利润往往只被少数寡头所瓜分，虽然开发者创业蔚然成风，然而据艾媒咨询统计显示：2011年中国手机应用开发者实现盈利的仅占13.7%，主要以依附企业本身（如腾讯公司的开发者）或者个人研发应用产品依靠广告获利为主；亏损的占64.5%，持平的为20.8%。目前国内市场最恐怖的事有两件：首先，和上半年相比融资的困难程度陡增；其次，国内市场基本成熟，互联网巨头们已经跃跃欲试，这对创业公司来说杀伤力更大。曾经的蓝海已经变得危机四伏，红海时代来临。

## 拥挤的单行道

国内大多数开发者似乎都走上了一条单行道。

**游戏成为第一选择。**在App Store的Top10下载排行中，娱乐类和基础类应用的上榜比例基本六四开。收入最高的前30款应用中，也有20款都是游戏。开发者之所以偏爱做游戏的原因很简单：游戏用户的付费意愿更高。

**积累用户数成为首要目标。**既然国内市场，尤其是Android市场难以让用户为应用埋单，那么开发者只好将扩大用户数作为首要目标。

**广告为主要收入来源。**虽然用户数庞大，但产品的盈利手段却很单一。对大量开发者而言，广告还是目前唯一能够获利的方式。

**同质化严重。**由于渴望成功，而缺乏创新，所以山寨成功案例成为很多开发团队的首选方案。由于难以推广，且同质化严重，所以很多产品极难在应用市场上获得曝光率，从而死在了沙滩上。

这样一条顺应大势的道路，正在成为一条死路。

## 创业成本凸显

与传统互联网相比，移动互联网创业的优势之一就是其创业成本较低，然而目前移动互联网的创业成本也在不断增加。

**人才成本。**随着移动浪潮大热，一时间，涌入移动互联网的开发者数量众多，却良莠不齐。毕竟已长时间从事移动应用开发的人占少数，大多数开发者的互联网经历尚浅，使得整个移动互联网行业的优秀人才积淀不足。而对于小团队，人才薪资成本较高，同时还面临着巨头挖角和员工辞职创业的风险。

**推广成本。**由于越来越多的人投身于移动互联网创业中，所以国内的移动应用数量陡增。虽然应用数量上去了，但同质化极为严重。一款好的产品面市后，往往会遭到各种形式的复制。数量众多的同质化应用，成为了一堵应用墙，阻挡着新产品进入用户视线。

**运营成本。**先做数量、流量，再谈收入是很多开发团队普遍的思路。然而，用户量激增带来的问题就是运营成本增加，尤其是对于社交型、社区

型团队，如何保证用户品质化、内容品质化，是不得不考虑的问题。

## 生态系统紊乱

**糟糕的应用商店。**丁香园CTO冯大辉曾撰文抨击应用商店糟糕的用户体验，主要有以下六点：信息各有一套，产品的描述信息以及应用类别划分，每家都自成一套；版本控制问题，多数都没有相对靠谱的版本控制机制；后台可访问性，后台访问速度慢，上传接口也有很多细节问题需要改进；界面信息混乱，对于开发者来说，后台提示和术语难以理解，文案差异性太大；数据不准确，统计信息不准确，有些市场数天都不更新统计数据，甚至不做后台数据统计；流程复杂，开发者操作流程复杂，应用审核进度慢。

**混乱的付费市场。**开发者们似乎已经放弃让Android用户掏腰包购买商品。而在App Store，付费情况又让人大跌眼镜，很多开发者在结算时面对高得令人发指的坏账率颇感无奈。如今，苹果App Store支持人民币支付，但是黑卡问题主要牵扯到一些贪图便宜的用户和刷应用排名的开发商，所以这对黑卡问题不会有太大的影响。

**让人不知所措的流量。**目前，已经有很多应用获得了大量用户，积累了大量流量。本来流量是依靠广告盈利的有效保证。然而，有开发者透露，过多的流量无法转化为盈利，并且广告商们也不愿意为太多的流量添加运营成本。

## 取胜之匙

**知识产权。**移动时代与PC时代有太多的相似之处，一开始是平台之争，各种平台百花齐放，但开放平台支持用户DIY的Wintel联盟胜出，不过，不管风云如何变幻，知识产权始终是灵魂，平台之后必然是商业模式的争夺，知识产权就是这些纷繁争夺背后的一个至高点。

**创新。**如何突破亿万应用筑建的高墙？也许推广能够让你提高应用展现在用户面前的几率，但是这并不会长久。而仅仅依靠产品内容的创新，最

终只会被数以万计的复制品淹没。所以创新不应局限于产品内容的创新，更应该包括推广与盈利模式的创新、对用户需求把握的创新、甚至于团队内部构建和工作流程的创新。一个好的创意赋予了产品生命力，但整个团队的持续成长往往需要持之以恒的创新动力，只有将创新深入骨髓，才能坦然应对多变的市场环境。

**构建产品文化。**《愤怒的小鸟》之所以能长久地霸占App Store排行榜前几名的位置，其原因早已不能用其游戏的好坏来分析。诚然，不断的版本更新带给这款游戏持续的生命力，但其经久不衰的原因在于“小鸟”们早已跳出了游戏的概念，成为了一种文化理念。“小鸟”的周边产品早已霸占了我们生活的各个角落，在无意识下融入了生活。因此，“小鸟”们无法被人们遗忘，延续着其活跃的生命力。

**用户品质化。**对于社区类及社交类产品，庞大的用户数量带来的并不是大量的现金流，而是大量的运营工作。有人认为，良好的用户环境是依靠产品上线后的悉心经营，但实际上，在产品设计之初，就应该考虑到目标用户的性格、喜好等因素，从而在产品的体验到规则上都设置筛选机制。好产品并不等于用户量庞大的产品，往往具有“小众”气质的设计更能保留其核心价值。

**步步为营。**都说移动互联网时代是个“快”时代，但实际上，每个公司都应该保持自身节奏，而非急于求成。虽然先发优势在这个时代下有着无可比拟的作用，但一款不成熟的产品是极容易遭到复制并被竞争对手打败的。研发周期多久、何时切入市场、迭代周期怎样，这些问题都应该从团队特性和产品特性来衡量，切勿求快、求广。

## 结语

移动互联网的井喷时代，红海赤潮来袭，惊涛骇浪遮掩了没有先发优势的开发者们的视线。但只要拥有正确的方向，小心规避暗礁，就能持续向前。2011年的中国移动互联网，数以万记的开发者们扬帆远航，波澜壮阔的时代，就在前方。P



年度十大热词：平台 | 移动 | 创业 | 大数据 | 编程语言 | 乔布斯 | 用户体验 | 并购 | 创新 | 泡沫

# 创业

文 / 卢刚

领域热词：社会化、电子商务、云计算、手机游戏

2011年的中国互联网创业，有大笔的资金，有创业氛围浓厚的咖啡厅，有分布各地的孵化器和创业基地，有众多的天使投资人，更有许多充满激情的程序员创业者。很热闹，也很潇洒。

如果要为2011年中国互联网创业列出几个关键字，第一个想到的一定会是：移动互联网。移动互联网本不是一个新兴的市场，但2011年，谢谢中国3G网络的迅猛发展，iPhone/iPad继续改变着世界，Android手机的品牌化和价格平民化，移动应用商店模式的流行，移动互联网为众多创业者开辟了一个更新、更广阔也更开放的实践空间。

## 2011年创业热点回顾

**社会化：**微博的传播力让许多人再次看到社会化媒体的力量。各大微博的开放API为许多创业项目提供了获取和培育用户的捷径。同时社会化问答、社会化评论、社会化招聘等许多传统互联网服务形态都在努力整合进新的社会化元素。

**“签到”服务：**移动互联网离不开手机，也就离不了地理位置。你到了什么地方，“签到”留名是看似很自然的事。“签到”的好处不但可以认识到同样到过这里的其他用户，最重要的是你可能得到该地商家的某些优惠。作为LBS最基本的模式体现，“签到”成为2011年的创业热点。

**移动通信：**移动通信方式的革新首当其冲。当移动运营商们还在思考着什么才是3G网络服务中的杀手级应用时，Kik不但很好地利用了通讯录将用户最有效地联系起来，也将用户之间的通信方式从短信转化为数据流量中的消息。Talkbox更是将文字内容拓展为更直接的语音。国内的创业公司

从开始一些小团队的努力到米聊的一度风靡，再到目前腾讯微信的强劲势头，虽然现状让不少创业团队叹息，但Kik+Talkbox对手机通讯录的使用却为众多移动互联网创业者打开了思路。

**移动社交：**手机成为社会化天然的工具。相对于互联网中市场占有率和活跃度依然很大的传统意义上的社交网络（如QQ空间、人人等），移动社交更像一片处女地而成为众矢之的。虽然都是针对移动社交同时产品均以地理位置为核心，各个创业公司的切入点却不尽相同。或基于场景，或重视熟人网络，或偏重陌生人交友，有全开放式的，也有基于兴趣的封闭圈子。移动社交试水者虽多但尚未形成气候，而大佬们似乎也还没真正开始发力。

**电子商务：**社会化电子商务的概念已经开始被众多创业公司所尝试。诸如美丽说、蘑菇街等社会化电子商务应用结合移动应用都有很好的口碑；电子商务行业中虽然各位老大的地位没有太大的变化，但垂直化市场仍然存在很大的开拓空间，比如女性化妆品市场；2011年不少创业者也把电商放在了奢侈品市场上，他们在奢侈品国内外巨大的价格差异上做文章，利润空间较大的基础上也背负着较大的资金周转压力。当然，在移动互联网的大环境下，方便用户直接在移动设备上直接消费或者在他的周边找到消费点也是电商创业的一个重要出发点。

**图片：**从Web 1.0单纯的图片上传和到Web 2.0的图片分享，图片在互联网的多媒体世界里一直占有重要地位。而在移动互联网时代，图片更是成为用户分享内容中最重要的形式。Instagram作为国外移动图片应用的代表被国内众多创业者所关注，虽然盈利模式尚不清晰，但其9个月内1.5亿张图片的分享量足以让国内业者坚定地走下去；新

的比如类Pinterest的以图片为核心的分享和社交网站在年末也成为创业的热点，而图片的衍生行业如社会化打印服务也陆续涌现。

**云服务/云计算：**云服务/云计算不再是浮云。云服务的概念被更多的互联网用户接受或者使用，比如云存储服务。对于创业者来说，云服务也不在大公司才能玩的游戏，因为移动互联网，众多的移动应用实际都是瘦客户端而将各类数据放在“云”端，而类似Dropbox、EverNote的云存储和云同步服务也开始默默地教育市场。

**手机游戏：**《愤怒的小鸟》、《植物大战僵尸》等在2011年风靡全球，使手机游戏成为国内创业者的香饽饽。《二战风云》、《捕鱼达人》等为代表的中国制造游戏在收入和用户数上都有不错的成绩。

苹果App Store模式让运营商走下神坛，也给了开发者、创业者更开放的商业和发展空间。而中国互联网经过10年的沉淀，借助移动互联网的大环境，的确可以说在2011年再次迎来了创业热潮。

## 反思

然而我们必须看到，众多移动互联网的商业模式并不清晰，只赚吆喝不赚钱的应用比比皆是。有人说现在移动互联网大多都是在玩投资者的钱，不无道理。手机游戏和社交游戏中出现了一批收入可观的创业公司，然而他们大多仍把重心放在海外市场，国人的钱依然不好赚；国内市场的移动应用仍然需要依靠广告模式，但对于大部分用户数量级未到百万级应用，其单靠广告的收入仍然有限。

创业公司的产品同质化严重，而中国许多投资者们在移动互联网投资中也在跟着硅谷投资者们的风。他们押宝国内的对硅谷模式的复制者们，抢好一块地盘再说的心理依然占主导地位，这在一定程度上就是在遏制中国创新。

中国互联网的资本市场不缺钱。众多投资早期创业项目的美元和人民币基金在2011年纷纷建立，或来自知名VC，或来自政府，或来自民营企业，或来自重量级媒体。大家都已经意识到孵化器和早期投资作为创业生态系统中的重要一环在国内的缺失，但怎样才能更有效地激发创业团队的创

新意识仍然需要继续实践。

众多天使投资者的出现对整个行业是件大好事。但我们也应该看到，天使们的身份、背景、资历、实力等参差不齐。天使们让创业者们有时摸不着方向，而真正有创业经验的天使投资人仍然很少；除了资金，创业者们需要更好的导师。

2011年，中国互联网的创业生态环境改善很多，但仍难以称之100%健康。

## 展望2012

SoLoMo (Social, Local, Mobile) 和O2O (Online-to-Offline) 作为行业发展的必然趋势，仍然会是创业的重点。但SoLoMo如果做不好，路会越来越难走；做O2O必须得静下心来把自己熟悉的垂直领域做透。否则，2012年也许会是更多大佬发力移动互联网的一年，不要让他们断了自己的路。

移动支付很难是一般创业公司玩得起的市场。但我们在期望出现在移动支付上创业甚至创新公司的同时，更希望大佬们能够拿出更让人信服、更简单的移动支付解决方案。因为这是2012年移动电子商务能否真正飞跃的重要前提。

2011年我们关注更多的是内容的量。微博的海量信息、“签到”模式中的POI等，2012年期待有更多公司把精力放在大数据挖掘、处理和再使用上。因为移动应用服务归根结底首先是数据为王，因而如何帮助用户更好地发现和使用数据从而提交更多更好的新数据会是决定公司能否出类拔萃的关键。

2012年也许是NFC近场通信的元年。因为听说多家手机制造商都将在2012年推出嵌有NFC芯片的手机。如果是这样，那么创业的话题范围相信会拓展不少。增强现实在国外已经出现了很多商业应用，渗透在众多的行业中，与现有移动应用的结合也是很好的方向。

总之，移动互联网关系的就是人和生活。创业机会就在身边，让你所做的事实实实在在地去满足人们的需求，研究并解决好一个问题常常就是创业迈出的最好一步。

2011年中国的互联网创业，很热闹但缺乏亮点；2012年，平淡一些、多些理性相信会更好。P



年度十大热词：平台 | 移动 | 创业 | 大数据 | 编程语言 | 乔布斯 | 用户体验 | 并购 | 创新 | 泡沫

# 大数据

文 / 占超群（离哲）

领域热词：存储、传输、HBase、流式计算、实时计算、图形计算

IDC报告称2011年全球将产生1.9ZB（1ZB=10亿TB）的数据，并将2012年称之为“Big Data”年，可见大数据处理的重要性；目前Facebook、淘宝等一线互联网公司都已经进入数据化运营的时代。

大数据处理的一个主要目的是改进用户体验、发掘数据价值、指导业务运营，基于此目的我们从数据的传输、存储、计算、展现四个角度来盘点一下大型互联网公司2011年的大数据处理方式。

## 传输

在高并发持续产生大数据量的情况下，数据的收集、传输、装载也就成了一个首先要解决的问题，解决此类问题的常规做法是采用消息队列方式，按照主题/点对点的方式进行数据的分发和消费，这类消息系统对性能、实时性、顺序性、可靠性、可用性、吞吐量都有较高的要求。此类代表有LinkedIn的Kafka、淘宝的Timetunnel、Twitter的Kestrel、Facebook的Scribe、Hadoop的子项目Chukwa、Cloudera的Flume等。其中Timetunnel、Kafka都在2011年开源。

## 存储

大数据处理的一个难点就是如何把大量的数据存储和组织起来，便于快速读写，由于数据量过大，已经远远超过单机和常规的磁盘阵列的存储能力，因此各种分布式存储系统应运而生。

**列存储数据库领域：**列存储数据库具有读取数据量少、减少重复数据、更适合压缩的特点，在海量数据分析领域得到了广泛使用，首当其冲的就是HBase。HBase是一个高可靠性、高性能、面向

列、可伸缩的分布式存储系统。随着Facebook、Salesforce.com、淘宝等大公司在HBase开源社区上的持续投入，2011年HBase发布了5个版本，在可用性和稳定性上得到了较大改善，并成为一线互联网公司首选的存储系统。目前Facebook的“Messages”应用就是基于HBase的，每天处理的消息数超过80亿条，以及超过750亿行数据的读写操作。

另外Hypertable等列数据存储系统在一些互联网企业也得到了应用。

**K/V型数据库领域：**K/V型数据库由于数据结构相对简单、接口简洁，在读写方面具有极高的性能，也得到了广泛的使用。

2011年，Cassandra、Riak、Voldemort、Tair等K/V型分布式存储系统在Twitter、LinkedIn、淘宝等公司得到了大规模应用，值得重点关注的是作为创建并维护Cassandra的Facebook于2011年彻底抛弃了Cassandra，全面拥抱HBase。

**文档数据库领域：**文档数据库有丰富的数据模型，数据格式自由，具有一个自包含的数据单元，支持多版本，性能相对较高，也受到不少企业的欢迎。

2011年，MongoDB、CouchDB等文档型数据库在很多互联网企业有了一些应用。如Craigslist、Foursquare就用MongoDB来做分布式数据存储。

**图形数据库领域：**图形数据库善于处理大量复杂、互连接、低结构化的数据，这些数据变化迅速，需要频繁的复杂关系查询。

图形数据库领域的佼佼者Neo4j，是一个嵌入式、支持完整事务的、分布式图形存储引擎，在2011年进行了三个版本升级，在稳定性、多语言支

持、图形计算完整性上进行大幅度完善。此外还有谷歌的Pregel、Twitter的FlockDB。

**其他：**基于常规数据库进行分库、分区、分表搭建的集群系统也有广泛应用，如Facebook、淘宝就有多个基于MySQL数据库搭建的数据库集群项目。

基于特定场景进行定制开发的分布式存储系统也有很多，如淘宝2011年开源的OceanBase具有高一致性、高可用性，并且支持分布式事务。

## 计算

海量数据计算主要有以下几种常用场景。

**流式计算：**因为数据的价值随着获取数据分析结果的时间成指数级下降，为了解决数据的实时获取问题，流式计算今年成为了热点。流式计算指为了达到实时性的要求，数据以管道的形式，源源不断地输入到流式计算系统，流式计算系统按照预定规则完成CEP（复杂事件处理）、简单计算分析等需求。

此类项目2011年非常多，有Yahoo!的S4、Twitter的Storm、HStreaming、StreamBase、阿里巴巴的IProcess、淘宝的Galaxy/SuperMario/Beatles等，其中S4、Storm都已开源。

**实时计算：**实时计算指的是对根据查询需求从海量数据中实时进行排重、排名、汇总等运算，此场景主要是针对数据的维度不能穷举、无法预先建立索引（每一列都有可能成为索引）、计算规则复杂需要工作流并且有高实时性要求。

此类项目有淘宝的Prom、实时数据计算引擎等。

**图形计算：**随着目前各网站对用户关系、网页关系等数据的深度挖掘，大型的矩阵计算、图形计算、网络计算不断发展。这些运算都有相同的特点：规模超大，常常达到数十亿的顶点和上万亿的边，在其上进行高效计算是一个巨大的难题。解决此类问题需要引入MPI或BSP模型来进行规模化、高效的图形计算。

此类项目有Google的Pregel，还有仿照它而推出的Apache Hama开源项目。

**离线计算（持续计算/批处理）：**离线计算指可以采用批处理的方式，进行数据的各种复杂运算，

此类场景对时效性要求不高，一般都是采用Map/Reduce模型来设计。

此类项目有Facebook的Hive、Yahoo!的Pig、MAPR的MapR等。值得重点推荐的是UC Berkeley推出并开源的分布式计算框架Spark项目，它轻量、接口方便、高性能，封装了很多常用的算法，在Twitter、豆瓣网得到了应用。

**其他：**此外，也有一些针对特定场景的海量数据计算的项目，如Google用来处理需要与网络进行大规模交互数据集计算的Dremel、需要对大数据进行实时更新的Percolator；还有LinkedIn推出并开源的支持实时更新、分布式搜索的Senseidb项目等。

## 展现与交互

数据分析最终需要很好地呈现给业务人员以及反馈给其他业务系统，因此最终数据的元数据管理、安全授权、交互格式、查询方式和优化等需要着重考虑。

查询方式有基于OLAP的MDX语言、Hive SQL等，可以用R语言进行分布式分析的RHadoop更是将基于Hadoop的分布式计算易用性提到了一个新的高度。值得特别推荐的有eBay公司推出的qlio事件驱动语言，它松散依赖SQL和JSON，可以完成多请求合并和数据打包输出。

具有元数据管理、可视化IDE、性能优化和诊断功能的项目有淘宝的数据工场等。

## 总结

大数据处理产生的价值将深入影响到互联网企业的运营、运维，相关项目也越来越多，但没有银弹，需要结合企业自身情况做出选择。

因为随着业务模型的不同，一个企业可能有多个分布式数据处理系统，如果每个系统都需要去开发资源管理、监控、调度、Failover、scaleout等特性并不可取，建议选择一个通用的资源层调度管理框架。在此方面，2011年UC Berkeley开源的Mesos以及MAPR的Resources Manager都进行了不错的研究和尝试。P



年度十大热词：平台 | 移动 | 创业 | 大数据 | 编程语言 | 乔布斯 | 用户体验 | 并购 | 创新 | 泡沫

# 编程语言

文 / 高昂

领域热词：Objective-C、Dart、Ceylon、Lua

## 语言排行榜

从2011年12月TIOBE编程语言排行榜的统计数据来看，应用范围最广的编程语言依然是老牌的Java和C。季军C++与排在第四位的C#则差距不大，并且C#近期还一直保持着不错的增长势头，或许在新的一年里会很快超越C++的排名。

当然，TIOBE年度最佳编程语言无疑是最受人关注的奖项，这个奖项被认为是编程语言在该年度市场份额增长及用户认可度方面获得了最为突出的成绩。历年获奖的最佳编程语言包括2004年的PHP、2005年的Java、2006年的Ruby、2008年的C和2009年的Go语言，Python语言凭借易用的语法和在Web开发项目中的灵活运用，获得2007年和2010年两次年度最佳编程语言的殊荣。

从2011年编程语言统计趋势来看，在iPhone、iPad等iOS平台应用快速增长的带动下，目前TIOBE排名第五位的Objective-C相对2010年的使用率增长了3.56%，排在各编程语言年度增长率的首位，2011年度最佳编程语言奖项应非Objective-C莫属。

在C语言基础上添加面向对象拓展的Objective-C语言，已发展成为苹果手机、平板电脑和个人电脑等各iOS平台上应用开发的主力语言。在已有C语言代码、类库等资源重用方面，Objective-C具备先天优势，同时Objective-C将垃圾回收等一些高层次的语言特性与C语言低层级的函数功能相结合，通过高层次的语言功能提高程序员的生产力，便于开发者在iOS平台上完成各项应用的快速实现。在新的一年里，搭载号称世界最先进的移

动操作系统iOS 5的终端随着手机的更新换代必将越来越普及。增加了200多个用户功能和1500多个新API及开发工具的iOS 5，让Objective-C语言开发始终保持着火热的增长态势。众多开发者也紧跟苹果系统不断推陈出新的步伐，升级或推出更为丰富的应用。

## 新兴语言

在新兴编程语言方面，刚刚过去的2011年不乏亮点。Google于10月在GOTO技术开发大会上发布的Dart语言，无疑是年度最耀眼的黑马，从10月首次进入TIOBE榜单的100名开外，到12月排名一跃升至60余位。Google在开发者心目中的影响力可见一斑。Dart语言面向结构化Web编程而设计，是继目前排名34位的Go语言之后，Google推出的第二门编程语言。Dart应用于网页程序开发，在语法设计上尽可能简化，以便于开发者上手使用并减轻Web应用开发的工作量。Dart在面向对象编程结构、静态类型和较大型应用开发方面填补了Web编程的空白，为传统以JavaScript为主的Web编程带来大幅度性能改善。

从目前Web应用终端发展趋势来看，移动设备、平板电脑等不同平台、不同规格的显示终端为Web应用开发带来了更为复杂的兼容性测试需求，而Dart这样具备良好的平台独立性，依托浏览器展示给用户的开发语言，在一定程度上简化了应用开发者在现代浏览器、移动设备和平板电脑等不同显示终端下的测试工作量，自然在语言发布后的短短两个月内，吸引了不少开发者开始试用并以此开发Web游戏或应用。随着以Chrome

为代表的浏览器逐步加入原生的Dart代码运行支持,今后Dart语言应用的运行将不再依托浏览器内置的JavaScript引擎执行,这为Dart语言的运行效率带来了极大提升,更符合Dart语言轻快、高效的设计理念。

各位开发者是否还记得, Gavin King在2011年4月首次公开的新一代JVM语言Ceylon? 发布时, 因其在企业开发领域赶超Java的目标定位引起开发者关注。尽管后续一直没能出现有说服力的Ceylon应用, 但Ceylon语言却一直向着既定的目标稳步迈进。2011年末, Ceylon再一次更新了语言发展的线路图, 给出Ceylon 1.0版本五个里程碑的详细计划, 同时发布了Ceylon 1.1及其后续版本的规划。

同样运行在JVM之上的Ceylon更加侧重于解决企业级应用开发与设计过程中遇到的问题, 在保持代码良好可读性的同时, 进一步降低开发入门难度。例如改善了JavaEE开发传统上对于配置文件的依赖, 充分借鉴了Java语言本身在开放性、健壮性和安全性方面的优势并避开其不足。特别是在生产效率 and 给程序员带来的工作负荷方面, Ceylon开发团队将投入精力进行改良。

作为一门深受Java语言影响的JVM语言, Ceylon由熟知Java语言的专家设计。尽管从头开始创建一门编程语言以及SDK工具包是一项耗费巨大工作量的任务, 但这项工程由Red Hat牵头并在开发组中汇聚了众多开发好手, 让不少Java开发者对其充满了期待。2012年, 随着Ceylon各个版本里程碑的如期到来, 是否会出现用Ceylon语言设计的杀手级开发框架或是成功案例, 帮助Ceylon语言从实验走向成熟, 成为真正有潜力赶超Java的企业级开发语言? 非常期待。

## 脚本语言

一直处于小众使用的轻量级脚本语言Lua, 在2011年展示出了值得称道的亮点和突破。6月, Lua语言首次进入TIOBE编程语言排行榜的前十名。这样的成绩无疑与iOS平台移动应用开发的火热, 特别是与移动平台的游戏开发所能带来的巨大商机密不可分。

Lua语法简洁优美, 解释器由标准C编写。拥有目前所有脚本引擎中最快的运行速度, 这样的特性让Lua成为用作嵌入式脚本的最佳选择。Lua代码几乎能在所有操作系统和移动终端上编译运行, 高效的运行效率让Lua在关注娱乐和用户体验的iOS平台上大有用武之地。

---

随着iOS系统热度升温, Objective-C排在2011年各编程语言年度增长率的首位, 2011年度最佳编程语言奖项应非它莫属。

---

Lua并非为构建完整的应用而设计, 其目的是嵌入在程序中为应用提供灵活的扩展和定制功能。在语言兼容性方面, Lua与游戏开发中广泛应用的C和C++有良好的互动性, 即C/C++代码可以与Lua脚本进行方便的相互调用。由于这样独特的定位和语言优势, Lua的最佳应用实践是在游戏开发中作为处理逻辑的脚本, 即游戏关卡的任务脚本。在以《魔兽世界》为代表的网络游戏中, 大量插件都在使用解释性的Lua脚本编写, 此外包括著名的游戏《愤怒的小鸟》在内, 不少iOS应用都大量使用Lua来加快游戏关卡的开发进度。

2011年末, Lua终于发布了让开发者等待数年的5.2正式版。已经在游戏应用程序开发中有着一定基础的Lua, 在新的一年里是否能够开拓更为丰富的应用领域, 将与其语言本身的发展及其终端平台的技术进步与市场占有率息息相关。但不可否认的是, Lua依靠其出色的性能表现和易用性站稳了脚跟, 势将成为移动开发语言的新宠。

## 结语

优秀的编程语言在应用开发中得到程序员的认可, 不仅要依靠自身优良的设计, 拥有一款先进易用的IDE以及活跃的开发社区也是语言在发展壮大过程中必不可少的支撑要素。2011年, 不少新兴的编程语言带给开发者耳目一新的感觉, 这让我们在新的一年里充满了期待, 期待涌现出更多更优秀的编程语言和开发框架, 让我们体验更为愉悦的编程开发之旅。P



年度十大热词： 平台 | 移动 | 创业 | 大数据 | 编程语言 | 乔布斯 | 用户体验 | 并购 | 创新 | 泡沫

# 乔布斯

文 / 张宁

关键词：苹果、死亡、App Store、iPhone、iPad

简单的在谷歌上搜一下“乔布斯”三个字，在短短的0.23秒内能找到22,900,000条结果。当你读到这篇文章，似乎会觉得它像是往大海中加一桶水那样显得多余。但是我们又不得不承认一点，无论哪个年度十大热词的榜单，必然会有“乔布斯”三个字位列其中。

## 逝世

2011年10月5日，一个人的死亡突然成为大家谈论的焦点。在很多关于乔布斯报道的文章中，我们经常看到这样的字眼：“我们失去了他，苹果失去了他，世界失去了他。”只有当那个“乔布斯”走了之后，我们才发现，有一部分永远也失去了。那个东西是在我身上，构成我的一部分。

在这样的感受里面，我们都经历了某种人生的转化。可以说哀悼不是一个个人的事件。它是社会的，甚至是以整个人类社群为基础的。正是在这种哀伤哀叹之中，我们发现了许多乔布斯的经典语录，发现了他许多优秀的特质。他也成为了我们每一个人学习的榜样。

## 战略大师

在乔布斯展现出非凡成就之前，早期许多观察家都低估了他掌控大企业、战略大方向的能力。1976年乔布斯和朋友成立苹果电脑公司。1985年在来自各方的压力下，乔布斯离开了苹果。1996年他带着NeXT最新的软件技术重回苹果掌控大

局。乔布斯在1986年创办的皮克斯公司，于2006年被迪士尼以76亿美元收购。1996年乔布斯回到濒临破产的苹果时，利用自己的私交搞到了1.5亿的救命钱。2005年乔布斯放弃了相伴Macintosh长达20年之久的摩托罗拉处理器，转而投向Intel的怀抱。这让Mac平台拥有了与Windows PC平台相同的性能表现。

乔布斯的每一次重大抉择，都要面临巨大压力。每一次的力挽狂澜，都是谋略的结晶。他确实是一名战略大师，尽管在早期就展现出了如此惊人的果敢，但是直到他去世后，他的这些伟大战略意图才慢慢被媒体和分析家们拼凑出一张清晰完整的蓝图。

## 从PC到消费电子

从一开始，乔布斯就希望制造出优雅而简单的苹果电脑，希望它们像是使用打字机甚至是烤面包机一样简单。在很早以前，微软就推出了PDA类产品，他们可以运行应用程序、播放音频和视频，处理文本。但是苹果将PDA、数字媒体播放器、手机、电脑都融合到了一起，很显然iPhone和iPad更受消费者的青睐。他是改变世界的天才，凭敏锐的触觉和过人的智慧，勇于变革，不断创新，引领全球资讯科技和电子产品的潮流。把电脑和电子产品变得简约化、平民化，让曾经是昂贵稀罕的电子产品变为现代人生活的一部分。

乔布斯和苹果则在不断突破产品设计，在功能和美学方面找到完美的契合点。他的创造性与打造

完美产品的企图心，让苹果和消费者有了双赢的局面。苹果得到了更丰厚的利润，而消费者得到了可以愉悦审美的产品。这更使得我们很难想象，如果整个苹果的背后没有乔布斯在掌控全局，苹果很难以这种制造完美产品的理念崛起。

## 遗产

乔布斯将产品研发放在最重要的地位，这是苹果取胜的真正原因。乔布斯开启了移动互联网时代的序幕，让智能手机、平板电脑、应用商店、云计算、版权数字内容分发和手势触摸操作等概念技术在大众中普及。

乔布斯将苹果帝国留给了老朋友库克，在未来库克能否秉承乔帮主的衣钵，将苹果再推到一个新的高度？事实上大多数人是喜忧参半，而这种不确定性也让更多的人去关注苹果未来的一举一动。

当死亡的阴影笼罩着乔布斯，当振兴苹果的伟大梦想激励着乔布斯，如果对自己所克服的巨大困难，毫不吹嘘和张扬，这就会产生出一种我们被称为优雅的美感。乔布斯举重若轻的优雅同样激励着我们每一个人，去面对生活的重担。

## 乔布斯经典语录

### 关于产品设计

“创意来自于我们对1000种东西说不，从而确保我们不会误入歧途或陷入太多琐碎的事务。我们总是在思考我们能够进入的新市场，但是只有学会拒绝你才能集中精力关注真正重要的事情。”

“这就是我的秘诀——专注和简单。简单比复杂更难，你必须努力让你的想法变得清晰明了，让它变得简单。但是，到最后，你会发现它值得你去。因为一旦你做到了简单，你就能搬动大山。”

“我们把屏幕上的按键设计得如此完美，就是为了让你情不自禁地点击它们。”

### 关于死亡

“记住，每个人都会很快死去。所有外部的期望、骄傲、对尴尬或失败的恐惧，在死亡面前，这一切




都会消失殆尽，只留下真正重要的东西。永远记住你将会死去，这一点将能帮助你跌入患得患失的陷阱。你本来就是赤条条来到这个世界上的，你也将赤条条地离开这个世界。想到这里，你也就没有什么理由不遵从你内心的想法了。”

### 关于工作

“成为坟墓中最有钱的人，对我来说毫无意义；晚上上床睡觉前能够说声我们已漂亮地干完活儿，这对于我来说妙不可言。”

### 关于创新

“创新来自于人们走廊上的交流，或午夜的电话，或意识到自己思维方式的漏洞。创新来自于某个人临时召集的六人会议，他认为自己明白了世界上最酷最新的事物，而且想知道其他人如何看待。” 



年度十大热词： 平台 | 移动 | 创业 | 大数据 | 编程语言 | 乔布斯 | 用户体验 | 并购 | 创新 | 泡沫

# 用户体验

文 / 董建明

领域热词：移动化、社交化、跨学科、核心竞争力

刚刚过去的2011年，可以说是用户体验行业飞速发展的高潮之年。这个趋势可以体现在业界对用户体验的重视程度、用户体验从业人员能力的提升等诸多方面。与此同时，这个行业也面临着如何把握机遇、充分发挥对产品影响力等挑战。

## 现状

在工业界，越来越多的人，包括高层管理者，更加重视用户体验。苹果等公司一次又一次的产品

成功，在很大程度上用事实证明了用户体验的价值。在信息化高度发达、产品同质化日益增强的环境下，一个成功的产品在功能和质量等传统指标的竞争力已经成为基本要求，决定产品成功的因素更多地取决于产品的设计、配置以及营销策略等是否能够最大程度地满足用户使用的各种需求。这些需求可能是效率和准确性，也可能包括易用性、趣味性甚至刺激性。对这些维度的把握涉及到社会学、心理学、市场学等人文范畴，远远超出了传统技术的关注范围。

随着对用户体验意识的提高，行业对用户体验的投入已经明显增大。在国内若干有影响力的公司，包括华为、联想、腾讯、百度、阿里巴巴等，都已经拥有数百人的用户体验团队。这些团队基本形成了用户研究、交互和视觉设计等系统的分工，并且建立了与产品相结合的管理机制。大量公司也在积极网罗专业人才，招聘非常活跃。

用户体验从业人员的能力结构和培养机制也在近期有了长足的进步。用户体验在十年以前的中国是一个鲜为人知的词汇，而现在国内已经拥有总数超过万人的专业从业者，并且这个数字还在迅速增长。比较资深的用户体验部门负责人往往也具有7-8年的从业经验。值得一提得多所高等院校也都设置了与用户体验相关的专业和课程，学科设置趋于完整，学校的学生对基本方法论和基本设计技能的掌握水平也逐年提高。很多学生在毕业前都有机会通过到公司实习和参加项目竞赛



本图来源于支付宝UED博文《我心中的用户体验设计》

等方式获取用户体验研究和设计的经验。

## 挑战

用户体验行业发展迅速，同时也充满挑战。这些挑战主要来源于信息和技术平台的扁平化。纵观各种企业的成功路径，我们可以看到两类不同的理念和操作模式。一类公司以产品和业务创新为驱动力，具有比较独立完整的研发体系和长期投入。这类企业居于少数，但它们是各个行业的领头羊，在激烈的竞争环境下能够占据主动，获得最大的收益。另一类公司是跟随型的。它们在基础研发方面投入较低，主要是利用扁平的信息平台模仿成功产品，作为后来者争夺市场。虽然这两种公司都可以成功，但跟随型的公司由于不具备先发优势，都会不可避免地面临强烈的同质化竞争，而要摆脱这种状况就必须向创新型的公司转化。

现在信息技术的平台已经向移动化、社交化方向发展，竞争格局趋于扁平。以前专属于专业公司才能开发的应用可以由很小规模的个体开发团队自主完成。以前只是由主流媒体控制的信息渠道越来越多地被微博等社交平台所占据。用户对内容的选择日益多样，对产品和服务的用户体验要求也随之提高。这些新的要求已经从消费领域逐渐延伸到办公领域，对传统企业应用的用户体验设计也设定了新的标杆。出色的用户体验不仅能增加产品使用过程的舒适程度，而且能正向引导用户对产品的使用方式，提高公司产品的黏性和品牌的认可度。这些都是对公司核心价值的直接帮助。相反，比较差的用户体验设计直接造成用户流失，影响整体商业目标的达成。

## 用户体验牵引产品

基于上述宏观趋势，有些领先的公司已经从总体策略上提出了“用户体验牵引产品”的概念。这个概念的基本含义是首先通过了解用户需求，勾画用户体验的愿景，再围绕愿景去规划和设计产品。要做到这一点不仅需要对产品特性进行分析，更需要对行业趋势和用户深层的心理模型进

行深入的挖掘。在这些基础研究方面，欧美是比较成熟的。大公司每年都有固定的投入去跟踪业界动态，预测未来方向，也会发布各种研究的报告。做这些报告对专业性的要求非常高，在提高专业性方面，我们国内用户体验行业尚有明显的差距。

## 如何提高影响力

用户体验行业近期最关心的问题之一是如何提高影响力。我们很多主流企业的文化还是过分强调技术导向，对于用户体验的理解还停留在视觉的美观程度等表面层次。这样只能在微创新的水平上止步不前，无法做出在用户体验方面有明显创新的产品。用户体验必须由企业高层的直接推动和支持，并且应当以长期的、策略型的眼光去进行持续投入，成为产品的核心竞争力。要得到高层的支持，用户体验的从业人员也要不断提高自身工作的专业性，还需要有较强的沟通能力。用户体验为中心的理念只有被公司广泛接受并融入流程，形成系统的创新文化，才能使得用户体验的价值达到最大化。

## 结语

用户体验行业，作为在中国的一个新兴行业，在当今众多企业强调用户体验的环境下，拥有最佳的发展机遇，潜力无疑是巨大的。同时，这个行业的基础是跨学科的，有较强的软科学性质，流程没有标准化，结果也往往难以量化。因此，各个企业的用户体验工作的成效呈现两极分化的趋势。比较成功的用户体验部门已经做出了很多令人信服的案例，在企业中发挥越来越重要的作用；用户体验工作开展得不理想的部门则得不到重视，甚至作为非核心业务而边缘化。这些工作的成功很大程度上在于是否拥有丰富经验的跨学科人才的支撑和引导，所以人才的竞争将是未来一个相当长阶段内最稳定和持续的行业特征。P



年度十大热词：平台 | 移动 | 创业 | 大数据 | 编程语言 | 乔布斯 | 用户体验 | 并购 | 创新 | 泡沫

# 并购

文 / 刘明

热点事件：Google收购摩托罗拉移动、微软收购Skype、EA收购PopCap、人人网收购56网

在21世纪的第二个十年，IT领域进入新的纪元——由春秋入战国。所谓春秋，群雄争霸，各占山头；所谓战国，兼并四起，合纵连横。为人才，为技术，并购开始了。

## 国外并购事件

让我们先说说国外的并购事件。随着从2009年的金融危机中复苏，各大企业的并购活动在欧美国家逐渐活跃起来。移动和通信产业成为最耀眼的主角。

最惹人注目的要数Google以125亿美元收购摩托罗拉移动，此次交易被认为通过摩托罗拉移动的专利保护已经被专利官司烦恼得焦头烂额的Android手机系统。可以说Google在移动领域的并购一直很有策略性，想一想2005年Google收购仅成立22月的Android公司，不但得到了现在炙手可热的Android手机系统，还收获Andy Rubin这名大将。这次Google的野心可能更大，除了凭借这一交易与苹果iPhone有了谈判的筹码，结合摩托罗拉移动先进的电视机顶盒制造技艺，Google TV将会得到进一步的普及。

2011年在Google已经完成的交易当中，除了与Yahoo!建立搜索广告合作的交易未能获得通过之外，包括收购DoubleClick、AdMob的交易均已获得了批准。

另一起巨头间的收购也颇为引人注目：微软斥资85亿美元收购Skype。外界将这次非同寻常的高价收购的真实原因解读为，微软想把Skype植入

到Lync、Xbox Live和Windows Phone甚至下一代操作系统Windows 8中，并加强自身其他产品的能力。

高通31亿美元收购Atheros和CenturyLink以25亿美元收购Savvis也是发生在通信领域值得关注的并购事件。

除了通信和移动，企业软件、云计算领域也发生了大的整合，SAP以34亿美元收购美国云计算服务公司SuccessFactors，从而加强其在企业云计算市场的竞争。此前SAP在把企业软件业务转向云计算领域的过程一直非常缓慢，其绝大多数软件目前仍在企业昂贵的数据中心运行。SuccessFactors是一家以云计算为基础的公司，该公司的服务能够帮助企业追踪和评估员工。此次收购SuccessFactors，将可以加强SAP在云计算领域的竞争优势，SAP在声明中表示：“云战略是SAP未来增长的核心动力，SuccessFactors的领导团队和核心技术与SAP现有云战略优势产品的结合，有助于使SAP成为云计算领域的领导企业。”据悉，SAP预期本次交易会在2012年第一季度结束。

此外还有惠普以102亿美元收购英国软件公司Autonomy。惠普在2011年经过换帅的震动，企业战略发生了重大调整。惠普的硬件业务——特别是PC和打印机业务——从2010年开始便已陷入萧条，因此该公司正努力转型成为一家软件和服务公司。Autonomy开发的软件帮助企业管理和记录分发到整个企业的文档和数据。如果来得及调整元气，将有助于惠普实现企业的另一个发展阶段。

在游戏领域，EA宣布以6.5亿美元现金和1亿美元股票的价格，收购《植物大战僵尸》和《宝石迷阵》等游戏的开发商PopCap。收购协议还包括多年盈利支付，因此最终总收购价最高可达13亿美元。PopCap旗下的游戏已在全球安装了1.5亿次。10多亿美元的价格对任何公司来说都是巨大的负担，显然EA非常想得到PopCap。这次收购将帮助这家游戏巨头打入移动和社交游戏领域，与Zynga、Rovio和其他公司竞争。

创新性行业并购一直被欧美视为并购案件中的核心话题，管理手段钳制太紧，不利于行业创新，资源优化配置；太松又有一家独大的忧虑，用收购打压闷死竞争对手的案例也有发生。

以Google收购ITA为例，此家公司是美国旅游软件提供商，作为在线旅游媒介为客户提供定制的搜索引擎技术。Google一直以来都想进入旅游搜索市场，因此希望通过收购ITA为其打开通途。美国司法部担心，Google会通过此次收购，了解到该行业竞争对手的很多敏感信息，因此美国司法部要求Google和ITA开发并授权旅行软件，建立内部防火墙机制，并继续向产业软件研发注资。经过了层层监管，严格区分权益，当Google遵循了上述条件，才得到司法部的批准。

## 国内并购事件

国人常讲，分久必合，合久必分。分合本来平常，但在国内IT领域，惯性思维不会用收购解决问题，而是自己也做一个产品来参与竞争。所以这一年发生的收购事件会格外受到注目。下面我们分别详述。

人人网以8000万美元收购视频网站56网。2011年9月27日，人人公司对外宣布将以8000万美元全资收购56网。通过此次收购，56网将成为人人公司的全资子公司，56网联合创始人兼首席执行官周娟也加入人人公司管理层担任副总裁。这也是人人公司纽交所上市以来，首次重大战略收购。

华友世纪宣布与视频网站酷6网股权合并。2011年11月27日，华友世纪宣布和酷6网进行股权合并。合并后，酷6网将成为华友控股集团全资子公司



并继续保留其品牌。

华为5.3亿美元全资并购华赛。2011年11月15日，华为正式宣布出资5.3亿美元购买华为赛门铁克合资公司中赛门铁克49%股份，如收购达成，将使华赛成为其旗下全资子公司。这个新闻再次将低调的华为推到了舆论中心，也给云计算在中国的发展增添了推力。

华为收购华赛的答案是显而易见的，华为正在推进企业业务市场，云计算已经成为华为的重要发展战略，按照此前约定的“对公司进行进一步的股权调整”达到彻底整合华赛的目标，可以说是为华为的云战略搭建了更为完整的一体化解决方案平台。

此外，百度收购了创新工场旗下的魔图精灵团队，但未透露收购金额等细节。对这次收购，百度CEO李彦宏表示“未来两、三年内，图片会成就一批不错的公司。”

回顾一年来国内IT领域的并购，让人感到头疼的是大多是资本重组（并非说这样没有益处），很少有让人激动的、因技术和人才引起的收购。是的，在国内，优质的整合资源、促进行业优化发展的收购如此弥足珍贵。📌



年度十大热词：平台 | 移动 | 创业 | 大数据 | 编程语言 | 乔布斯 | 用户体验 | 并购 | 创新 | 泡沫

# 创新

文 / 刘明

“创新”是IT领域永不过时的buzz word。那么，回顾2011年，当我们谈论创新时我们在谈论什么？

## 狂飙的搜索词

新的产品，特别是大公司推出的，总是能吸引更多的关注与讨论。“创新”，是对它们最高的褒奖。Google发布了2011年度全球热门搜索关键字排行榜Google Zeitgeist。在全球热词榜单上，有三个IT名词闯入前十，它们是Battlefield 3、iPhone 4S和iPhone 5。

Battlefield（战地风云）是EA推出的一系列第一人称射击游戏。从第二代开始，每一版本在保持既有风格的同时，都较之前有些许创新元素的增加，比如改进渲染引擎、增加新的游戏元素和剧情模式、扩展新的平台等。

Battlefield 3是Battlefield系列的第十一款，采用新一代的寒霜2游戏引擎，是此款游戏的创新重点。寒霜2完全利用DirectX 11 API和Shader Model 5以及64位元性能，同时将包含增强的Destruction 3.0（可实现仿真的物理系统效果）。据称，未来还将推出Wii U的版本。在渐进的创新中，Battlefield 3销量首周突破500万，一个月后达到800套。

iPhone 4S和iPhone 5，一个已经行销全球，一个却还只存在于讨论的话题中。在iPhone 4S真相大白之前，人们都用iPhone 5代指Apple的新一代产品，并给了它极高的期许。在众多言之凿凿的预言中，iPhone 5将带来如下的创新改变：颠覆性外观（极为轻薄、弧形铝背的工艺）、裸眼3D、无按

键设计（取消Home键）以及超过人眼分辨率的显示屏等。然而世人看到的却是Tim Cook演示的iPhone 4S，外观与前代几乎相同。一片大呼失望声过后，人们认真点评了Siri的智能语音识别、新iOS、更稳定的天线设计等数百项iPhone 4S的软硬件新增与改进，可是普罗大众仍难免遗憾。是啊，创新如此艰难，连以此安身立命的Apple都难以回回有新招。可我们应当注意到，每一次Apple新产品的推出，都将行业的最高标准又向前推进了一大截。iPhone 4S如此，iPad2如此。

在中国的Google Zeitgeist榜单上，一款今年新推出的产品名列其中，那就是腾讯推出的微信。这款语音通信产品结合腾讯自身推广优势，在推出的极短时间内横扫国内市场。在微信之前，就有Kik、Talkbox等金牌产品在前，因此微信刚一推出就被指抄袭、山寨，几乎一边倒的负面评价。然而腾讯做产品的韧度在此处表现无遗，一方面优化产品性能和用户体验，一方面结合本土需要，增加LBS的摇一摇功能和视频通话功能，并且在iOS、Android和Symbian多平台推广，获得广泛的用户接受。在产品创新的概念上，腾讯自称的“微创新”也许只是说辞，但激发用户需求、以渐进式的功能添加来满足用户需求，是腾讯发挥自己创造力的优势项目。同样的，我们认为创新如此艰难，当独立来看微信这款产品时，它已经做得足够好。当我们谈到“创新”这个词，总是希望它代表的产品能够让人眼前一亮，绝无耳

闻。然而事实上，在相当理想的情况下，创新与山寨的界限也模糊难辨。

## 对创新的渴望

对创新的强烈渴望，往往来自残酷的竞争压力，“不创新则灭亡”的告诫，一直高悬头顶，让人不敢懈怠。

微软推出的Windows Phone 7和Windows 8创新性的UI设计Metro，在2011年一直被人津津乐道。究其创新动力，在Windows Phone 7方面，iOS和Android平台在移动市场日益壮大，面对不断增长的移动用户数量和甚至不断被侵蚀的桌面市场，这是微软为冲进智能手机市场不得不做的改变。但有Android多次被诉专利侵权的警鉴在前，甚至连设计元素雷同都会引来官司缠身，创新是微软不得不走的一条路。事实的反馈也证明这是极为正确的，也给微软在移动领域形成鼎立之势带来了极好的契机。在Windows 8方面也是如此，平板电脑的迅速风靡，让微软看到了未来的隐忧，于是连通整个产品线的创新产生了。这里我们也看到了创新不仅是公司某一项产品的问题，它会影响整个公司的战略发展。

与传统产业不创新尚可以坐吃山空几年不同，互联网行业往往需要不断推出新的产品，才能维持自身的关注度和市值。Facebook平台上的Zynga，推出游戏的速度非常之快，因为每一款游戏都会有用户疲劳的阈值（想想国内早被冷落的偷菜），必须推出新游戏才能吸引用户的兴趣。

开疆扩土到另外的领域，后来者往往需要有创新的产品才能抢得先来者的份额。屡次进入社交平台领域的Google就是这方面的例证，Google+不同于其他产品的圈子设计，统一的全新UI设计都表明了搜索巨人的决心。但Google之前的多次努力，创新度不可谓不高，Google Wave中就有许多前卫的技术应用，但多次的关停表明，仅有技术的创新，不能很好地了解用户需求，这样的创新只能是空中楼阁，看着高端，实则无用。

在2011年实现跨界发展的还有一个大公司——Amazon，它推出了传闻已久的自家的平板电脑

Kindle Fire。虽然在外界的性能评测上，Kindle Fire远没有那么出色，但赔本赚吆喝的价格以及与Amazon自身海量数字内容的紧密结合，使得Kindle Fire还是被众人认为是苹果iPad众多竞争对手中最为有力的一位。其实Amazon这次跨界出击是在时局逼迫之下的已成定局，苹果的App Store成为最大的电子音乐发行商、最大软件网上商店以及最大的电影电视节目租赁商，这一切都让以贩卖内容为主的Amazon尤为感到压力紧逼。在公司老大Bezos巨细靡遗的掌控下，Amazon以其强大的营销渠道打开了自己的销售平台。Amazon是不是在做创新呢？在苹果这个硬件厂商卖内容卖得热火朝天时，Amazon反其道而行，证明了卖内容的商家也可以做出自己的硬件平台。

## 我们的创新

我们也在谈论创新，似乎总摸不到门道。无意苛责国内做IT产品的公司，它们在推出一款产品时常常背负“山寨、抄袭”的批评，然而我们知道在没有完善、良好的知识版权的保护业界态势的情况下，“创新”只是奢谈，一步登天的妄想。

同时，我们处于下一个互联网的热潮当中，在电子商务、移动互联网、社会化、娱乐游戏以及电子出版领域都大有可为，搭手望去是一片片蓝海，我们的创新机会比比皆是，也许当我们谈论创新时，作为圈内人从自我做起，创新的勇气是第一位的。

## 结语

创新是保持IT企业青春的灵丹妙药，没有创新的公司是无法真正延续发展的。在一片喧嚣扰扰的2011年的创新之声中，我们至少可以做出一个结论：做出好的创新产品的公司，的确得到不错的回报。P



年度十大热词：平台 | 移动 | 创业 | 大数据 | 编程语言 | 乔布斯 | 用户体验 | 并购 | 创新 | 泡沫

# 泡沫

文 / 佳琦

提及率最高热词：云计算、电子商务、移动互联网、团购、SNS

“泡沫”入选2011年度关键词，源于可能大家都有一个很直接的感受：这个行业太浮躁了。很多领域短时间里聚集了太多说不清道理（来历）的人气，面子上非常繁荣兴旺，里子却或许隐含着虚妄不实。

## 泡沫堆起

泡沫也许源于一个不错的点子，也许起于极小一部分狂热爱好者的追捧。

云计算是个好东西，以此为基础，全球大大小小的互联网公司们衍生出许多新的创业模式，在节约能源、集约效率方面也都是未来的趋向。因此，云计算也被称为继个人计算机、互联网之后的第三次IT浪潮。与国外民间资本公司发起倡导、由下而上发展云计算不同，在国内，云计算主要由政府主导，已然上升为国家战略，被列为“十二五”期间国家重点扶持项目。因此很多地方政府都抛出了“云计算”的绣球，各种“云计算中心”相继落地。2011年，重庆“两江国际云计算中心”开建。随后京沪深四大一线城市的“云计算”项目纷纷上马，北京的“祥云计划”、上海的“云海计划”、广州的“天云计划”和深圳的“鲲鹏计划”，投入均过百亿。除西藏和新疆外，我国其余30多个省、市、自治区均提出或制订了云计算发展计划。

犹如跑马圈地，云计算在国内的发展态势让人心生隐忧。繁荣耶？泡沫耶？

称赞者说，国内中小企业数量巨大，他们对云计算的需求极为广泛，基础设施建设到位，引导他们逐步接受、充分利用资源，会使得多个行业联

动发展。云计算产业的原始积累，只有发挥聚集效应才能为其进一步发展带来保证。

唱衰者云，虽定义为战略性新兴产业，但在很多人眼中，云计算依然是第二个房地产，是无处可去的热钱的下一个目标。而云计算本身应用绿色驱动、节能等理念被选择性忽略，本来应该“软”绵绵的“云”如今变得很“硬”。

团购自诞生就受到追捧，其鼻祖Groupon创造了很多令所有人都惊讶的增长速度，一波三折的IPO甚至为国内网站撬开了上市窗口，但本家的日子也并不好过，利润无法维持高位的同时，还被指“庞氏骗局”，更别提它的众多效仿者。国内曾兴起千团大战的盛况已不复存在，有半数网站沦为难兄难弟，加之消费者和商家都已“团购疲劳”，也难怪2011年春天就有人预言团购要过冬。

然而，团购只是冰山一角，全中国的电子商务都有惴惴不安撞上轮船的担心。近两年VC对国内电商注资超过300亿，是个漂亮的U字曲线，然而电商们却在倒转的U字上急转直下。本身缺乏赢利能力，只能靠疯狂烧钱，砸广告、拉宣传、抢人才，“卯食辰粮”，年关将近却发现融到的钱“多乎哉？不多矣”。于是裁员减薪断臂求生，于是员工失控拉黑老板。挥霍无度，亦需索无度，每个人都相信干这个一定能赚钱，因为似乎总有更傻的人。

## 你看得到泡沫吗？

1994年，Mosaic浏览器出现，互联网诞生，新世界的图景在有才华的年轻人眼前平展无碍。他们坚信一种以互联网为基础的新商业模式将会兴起，

把才华、青春、热情投入到新一波技术浪潮中，于是WorldCom、Netscape、Yahoo和Google们，一个个IT巨人诞生了。

这样的场景正在重演——如今移动浪潮席卷而来。iPhone手机和App Store的成功，引发产业震动，大家都沿着苹果开创的成熟模式迈进，Google、微软、黑莓都开展了自己的应用商店，使出浑身解数吸引开发者。

这些开发者们同样是一群有才华、有抱负的年轻人，他们在手机上发觉了用户未曾想到的新应用、新需求。VC们目睹了智能手机用户的高速增长和更大利润空间的存在，故出手更快，一改以往的谨慎观望，先发占位就成了规定动作。策略是让众多竞争者进入，期待市场将对大回报率给予为数不多的胜者，于是他们给予了移动互联网领域高金额、高估值。一款新颖的移动产品出现时，往往受到广泛讨论与模仿。然而手机屏幕狭小容纳应用有限，用户对应用可分散的注意力有限，加之很多产品团队没有切实可行的商业计划和管理能力，却出于新颖的概念，仅为出售创意。一款产品急遽升温，获得巨额融资，转眼间就销声匿迹的事迹屡见不鲜。但前仆后继者众。

在国内，移动互联网喧嚣声在外，暗自琢磨下来却有不赚钱的尴尬。据统计，中国手机应用开发者近百万，亏损的占了六成，与家大业大的巨头相比，他们无力烧钱，或者说他们没有做好烧钱的准备。可能他们坚信依靠个人或小团队的努力也能放飞自己的疯狂小鸟，面对IT巨头们，小小的植物们也有战胜僵尸的可能。不过，有时候成功往往出于偶然，如果把偶然当作必然来相信的人太多了，泡沫将不可避免。

移动互联网尚在发展早期，二十多年前的壮丽景象已经显现，伴随而来的还有产业乱象和资本趋紧。在奋力一搏的同时，也要看到若隐若现的泡沫就在身边，毕竟十年前的互联网股鉴不远。

## 泡沫是有弹性的

常常看到有新闻标题有质疑，是否需要另一个Google？是

否需要另一个Facebook？就知道又一个身怀利刃的家伙闯入逼仄的战局之中。然而，凭借一招鲜吃遍天的时代似乎已经过去非常久远。一款产品能否脱颖而出凭借的是其母公司的集团作战、后援支持，以达到网络效应并形成行业垄断。

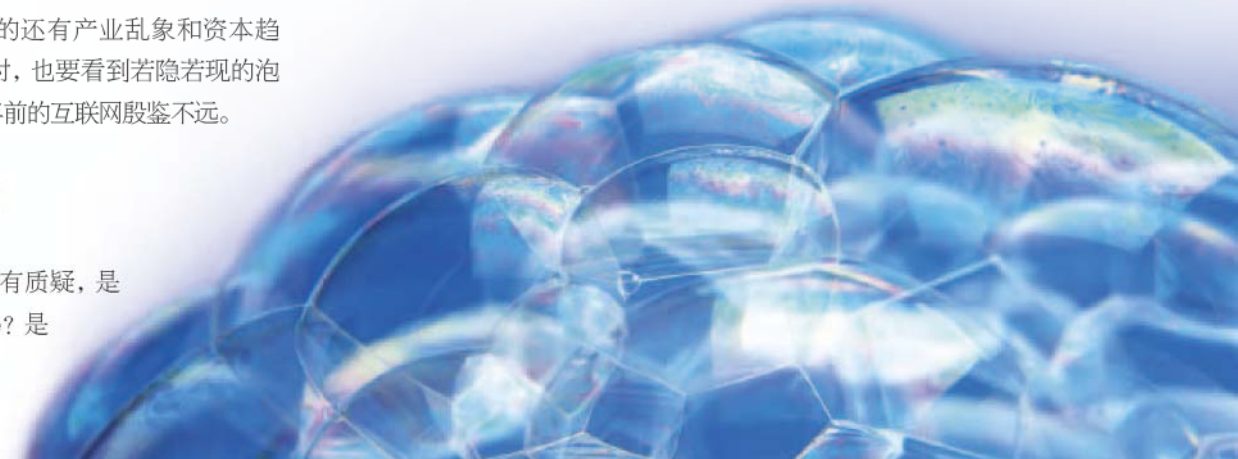
对于国内而言，每一个细分领域都有着大量公司存在，他们拿着仅有Logo不同的商业计划。很显然，无论想法多么与众不同，每一领域能被人提及的胜出者的名字只有一个，因而大部分公司会失败。

这里我们举几个例子。比如微博，这个产品已经成为各大门户的标配，然而无论多么卖力地宣传，能被记住并使用的产品寥寥。同样的在社交网站、LBS服务、社交游戏等领域，我们都看到一家和几家独大的趋势。然而仍有新的同类产品挑战既有秩序，除了完善自己的产品线和加速乱象外，泡沫的容纳能力也表现了出来。

我们知道，泡沫是聚在一起的许多小泡，这其中总有体积较大的泡泡散发着更五光十色的光芒，并将周围相形见绌的小泡泡融解吞并。在IT领域，道理亦然。

## 结语

17世纪，荷兰发生了郁金香狂热。后世对此解读的文献数不胜数，各家观点也不尽相同，但对于泡沫的基本特征描述确实一致的，即市场价格脱离实际价值大幅上涨。当一款产品获得没有足够用户的认可，但仍然受到吹捧，我们似乎就可以冷眼旁观看它挥霍完资产的燃尽图。无论是它们自己堆肥，还是击鼓传炮仗，我们知道，泡沫总有破灭的一天。P





# 年度十大人物（国外篇）

文 / 高博

## TOP 10榜单

Dennis Ritchie  
Doug Cutting

John McCarthy  
Ryan Dahl

Steve Jobs  
Drew Houston

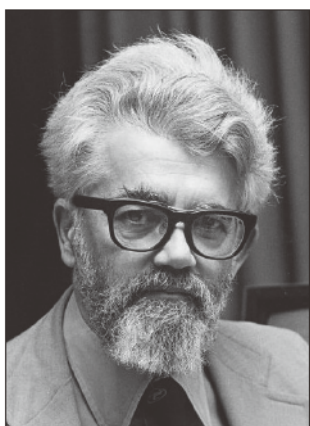
Paul Adams  
Yehuda Katz

Jeff Bezos  
Larry Page



### Dennis Ritchie

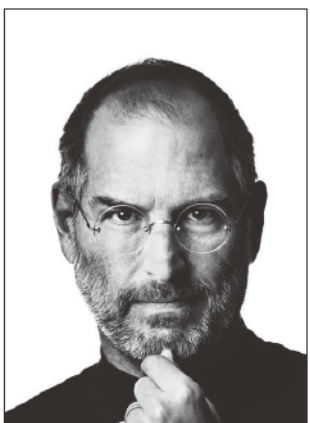
如果在Google上搜索Ritchie的简历，你会发现虽然有很多结果，但内容却相差无几。对于Ritchie的生平，我们能够确认的部分只有：他生于1941年9月9日，设计并发明了C语言和UNIX操作系统。Ritchie是一个彻底拒绝任何光环的人，连他的逝世，都是友人Rob Pike在2011年10月12日去其家中拜访时才发现的。但他的影响却渗透在我们的举手跬步之间，C语言是实现任何系统软件核心的唯一选择，而UNIX则是所有现代操作系统的原型和基础。C和UNIX都如Ritchie的为人一般，简约、明了、到处可用。Ritchie的著作《C程序设计语言》已经成为全世界C程序员无可争议的圣经，薄薄的200页书中，信息量却在几十年间满足了所有的编译器、操作系统、应用程序和系统架构的实现者和研究者。Ritchie已逝，人们却感觉他还活着，仍然带着那样慈祥的、与世无争的微笑，指导着信息世界前行的方向。



### John McCarthy

2011年，世界失去了太多智慧的头脑。而McCarthy的逝世，甚至让机器的智慧也在这一刻黯然。正是他，将人工智能注入了原本看上去只能完成机械计算的电子设备，而人工智能则是几乎所有电子游戏的算法基础。从这种意义上说，麦卡锡实际上是给这个世界带来了最多欢乐的人，也是使得电子设备与最广泛的大众娱乐相结合的根本推动力的促成者。

然而对于程序员来说，McCarthy给他们留下的另一笔宝贵财富则是LISP语言，以及函数式编程范式。按照当前的分类，LISP语言属于非纯函数式编程语言中的强动态类型，但受它的启发，像Haskell和Scala这样的强静态类型式的函数式编程语言已经渐成显学。甚至在JavaScript这样具备了语法潜能但多年沉寂的Web程序设计语言也因为这一点而重焕青春。



### Steve Jobs

大星陨落，万马齐喑，Jobs的逝去给世界留下了太多戛然而止的时代错位感。工业和商业的奇迹在Jobs的世界中，仿佛只是自然而然的家常便饭。但从Jobs的身上，却闻不到一丝技术男的迂腐气，更没有商人的铜臭气。他只是极力地将美学和禅意引入一个又一个影响着千千万万普通人的日常生活的产业中，造成一场又一场颠覆性的革命。即使是对手，也不得不折服于他对于极致和细节的追求。

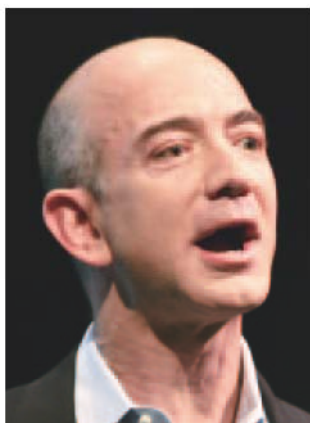
Jobs的一生，充满着不妥协，而这也正是深深地植根在苹果和皮克斯动画公司，以及它的每一件产品中的基因和血液。Jobs唤醒了人们心中已经麻木的那部分对于自由的主张，从而在太多人，尤其是创业者的心中取得了精神导师般的魔力，甚至被尊为“教主”和“神”。的确会有人不以为然，觉得单键鼠标不可理喻或是iPhone不过是个“加了手机功能的数码相机”，但即使是这些人也不得不面对这样的事实，那就是Jobs这个男人以一己之力单挑微软帝国，并且最终取得了胜利。虽然未来会是怎样，谁都说不清楚，但乔布斯却让所有人都提前看到了一些未来的影子。失去了Jobs的世界，注定需要在迷茫中探索更久。



### Paul Adams

社交网络已经是被炒烂了的主题，然而在其中仍然有推陈出新的灵魂人物。Paul Adams经过潜力研究，提出了细粒度控制社会关系的“圈子”概念，并成为Google+的核心研发理念。

Adams的核心思想在于，传统社交网络的核心是聚合，而事实上，这种聚合会带来隐私方面的困扰。Google+以及现在Adams所供职的Facebook正在发生的转变，则是把个人的整个社交网络划分为不同的“圈子”，每个人在不同的圈子中可以以不同的身份参与不同的活动，而圈子之间则是相对隔离的。比如，一个学校的老师可能同时在酒吧聚会的圈子中会发表一些尺度较为开放的照片，或对这样的照片发表评论，而这些则不会被他的同事圈子中的人看到。这就有效地保护了社交网络中的个人隐私问题，从而使得人们更愿意去使用社交网络产品，并向这些产品提供自己的真实状态和数据。Adams正在写他的新书《人以类聚》(Grouped)，让我们期待这本新作的问世。



### Jeff Bezos

Bezos成为新闻人物可以一直追溯到世纪之交亚马逊刚刚成立不久的时候，那是个电子商务概念还刚刚兴起，一切都还不明了的战国时代。如今，Bezos和他的亚马逊又成为了焦点，然而今非昔比，亚马逊已经是电商中的佼佼者，不仅拥有B2C和C2C的完整商业模式，也已经将业务拓展到了五大洲的每一个重要国家。

但电商领域的业务规模和商业模式方面的巨大成就，还只是Bezos成为十大人物之一的部分原因。亚马逊真正做到了与众不同的地方，在于它提供了自主知识产权的云计算平台，以及积累了全球最大的数字内容分发资源。亚马逊做云计算平台可以说并非偶然，电商业务所要求的海量交易请求处理，必然对基础的计算架构提出极为严苛的成本和性能要求。但Bezos仍然需要极大的胆识来做出自主研发云计算平台这样的决定，而且这个决心能够最终成为现实，其中所历经的艰辛和挫折也必然是外人所难以知晓的，更不必说这个平台还能够被抽象出来对外提供从基础结构到应用服务托管等各个层次上的云计算产品了。难怪，Bezos曾在不止一个地方说过“云计算平台是亚马逊最值钱的资产”！对于亚马逊发家的老本行——图书销售，它也没有原地踏步，而是敏锐地发掘了未来的读书形态：数字阅读，并斥巨资进行资源和战略布局。2011年，是亚马逊的收获之年，随着它的内容分发落地及设备Kindle的普及，它的数字内容销售也像芝麻开花节节高。Bezos和亚马逊的成功，是不是可以给每一个电商从业者，乃至每一个IT从业者提出一个课题：是永远做一个低买高卖、辛苦终日的搬运工，还是做一个提前布局、引导未来的弄潮儿？



### Doug Cutting

也许每个人今天听到Doug Cutting的名字，第一个反应都是Hadoop，但他更重要的身份是Apache软件基金会主席。无可否认，Hadoop已经取得了决定性的胜利，不仅获得了主要的Web产品和服务供应商Google、雅虎和微软将其作为自己的云计算数据中心的支持，而且连甲骨文也推出了自己建立在Hadoop基础之上的硬件产品。但Hadoop并不是空中楼阁，它是由搜索引擎这种建立在高度非结构化的数据和在海量数据中查询的需求作为原始驱动的，这个需求使用传统数据库技术很难满足，而Doug Cutting正是出身于搜索引擎核心研发领域。

也许，这揭示着未来的世界有多么丰富多彩，它可以由任意多项的属性来描述，并很容易地成为另一个更大数据集合的一部分。这听起来很像是宇宙本身，不是吗？人类的科技发展，也正在迅速地向这个方向迈进，这不就是几千年前老子说过的“道法自然”在数据处理领域的体现吗？





### Ryan Dahl

技术，很重要，但更重要的是想象力，而Node.js的开发者Ryan Dahl就是极富有想象力之人。首先，他统一了前端和后端的编程领域语言，把JavaScript语法引入后端开发中；其次，他采用了Chrome V8引擎，保证了解析的性能；第三，他在采纳了事件驱动的响应模型，尽可能地避免空转等待的同时，又抛弃了以存储器为中心的多线程模型。这些设计都充满取舍和权衡，都建立在对后端程序设计和服务器特点的充分掌握，和对JavaScript语言的优点也深得肯綮的基础之上的。

所以Node.js虽然架在Chrome V8之上却获得了微软的大力支持，而且成为LinkedIn和GitHub的移动后端，这都是完全不令人意外的事。当然，Node.js毕竟太过年轻，支持的框架还不多。但相信Dahl推出更有震撼力的作品，也是指日可待之事。



### Drew Houston

Dropbox公司凭借同名产品融得2.5亿美元，其CEO Drew Houston今年可谓春风得意。

不过，据这位老兄自己说，他当年面对Jobs数额更大的收购要约，却没有动心。而是潜下心来，做好基础服务，并将产品铺设在各种桌面和移动设备之上，在短短四年时间里用户数就达到4500万，终于获得了资本市场的肯定。可是他又有了全新的、野心勃勃的计划，Dropbox的目标是通过云存储服务来“简化人们的生活”，把文件寄放在云端，用户就随时可用自己的资源了。Drew Houston深知用户体验，尤其是非技术用户体验的重要性，他的典型客户就是不懂技术的投资商。

根据Drew Houston的经验，想在互联网上做出一番成就的话，小白用户们是一定要照顾好的呀。看起来，这一条经验，无论在国内还是国外都适用。



### Yehuda Katz

能者多劳。Yehuda刚刚发布了Rails 3.0，又马不停蹄地奔向下一个目标——HTML 5的JS框架SproutCore。作为jQuery的主要作者之一，Yehuda对于前端的熟悉程度不亚于后端。他的思维漂移到Web开发的整个架构中的，对他而言，中间那堵分隔着前端和后端的障壁，并不存在。

JS很好，Ruby很好，SQL很好，HTML5也很好，但都还不够好。好比从基础木料和钉子开始，如果不是高明的设计师，很难造出可用的家俱。而Yehuda的工作成果就像宜家的半成品，人们只需买回家，对照着图纸，就能很快地拼装好美观、耐用的家俱。然而把握软件工具的这个“中间层”，做出既灵活强大又适合一线工程师上手的框架来，是非常不容易的事。更何况是在开源社区中，统一认识更是难上加难。所以，Yehuda作为2011年十大人物，可以说是当之无愧的。



### Larry Page

当你开创了一家公司，有着四五百亿的现金，并且有八架专用飞机，你还要做什么？很多人在达到这个目标的千分之一甚至万分之一时，可能就会选择退休享受生活了。而Google的创始人，以及新上任的CEO Page却选择了“建立一个全新的管理团队”。

经过十年的管理实习后，Page重回大位，大刀阔斧地整顿高层管理团队，将原先的权力机构运营委员会重整为嫡系的“L团队”。新团队中除了核心产品Android、Youtube的领导人以外，还引入了财务和法务主管，而退出的成员则主要是人力资源和日常运营的官员，这表明Page明确地将用人和决策大权掌握在自己手里，并要开始Google新一轮的集中创新和发展步骤。果不其然，新官上任短短数月，就抛出了重磅产品Google+，传统核心产品搜索引擎和Gmail也换上了全新界面。

# CTO 俱乐部 CLUB

## 高级技术管理者大本营

CTO 俱乐部是专门为 IT 产业中的高级技术管理者打造的交流、分享平台，拥有超过一万名会员。会员除了可以线上交流外，还可以参加俱乐部定期在全国范围内举办的沙龙聚会、主题演讲等线下技术交流活动。

<http://cto.csdn.net>



CTO、技术总监、开发总监，立刻免费申请加入





# 年度十大人物（国内篇）

## TOP 10榜单

雷军 任正非 张小龙 吴刚 张宏江  
林仕鼎 云风 吴军 周娟 季逸超



雷军

入选理由：开创用互联网做硬件新模式，推出极富影响力的小米手机。

小米公司及其CEO雷军算得上是今年最具争议的话题之一，无论小米手机成功与否，“互联网公司做手机”这个模式也算得上是本年度国内最大胆的创新。

在雷军眼中，手机市场的竞争是软件、硬件、互联网的综合比拼，所以雷军将小米的模式称之为“铁人三项”。不仅如此，雷军还否定了自己之前在金山埋头自身发展的理念，实现自我变革，一开始就引入风险投资，依靠丰厚的待遇、期权等招聘了500名出色的工程师。虽然仍需市场检验，但小米手机的影响力已有目共睹，从组建团队到推广营销，雷军功不可没。



任正非

入选理由：突破长久以来的跟随战略，带领华为从电信服务商向终端和IT服务市场进军。

2011年，智能手机等消费终端成为华为最主要的驱动力之一，华为的终端设备部门员工已达1万人，华为终端的战略以自有品牌开始涉足大众消费市场，塑造华为终端在大众市场的品牌形象。任正非表示：“华为终端要成为这个领域重要的玩家，到2012年，销售额要超过100亿美元。”不仅如此，华为还想在云计算领域占据更大的份额，“请”来IBM帮助制定、实施品牌战略，空降IBM前大中华区云计算总经理朱近之等高级人才。华为之所以能保持正道直行，依靠的就是任正非的雄心、远见和决心。



张小龙

入选理由：率领团队打造了2011年最成功的移动互联网产品之一——微信，互联网Go Mobile的代表。

微信当前用户数已超3000万，而张小龙仍是一个谜，这位十年前就能称得上国内最“伟大”的程序员竟然很难被人在互联网上找到一张清晰的照片。但从Foxmail到QQ邮箱，再到微信，我们可以感受到张小龙对细节的追求、对事情极具探索的精神。虽然腾讯对产品的“微创新”一直被人诟病，但微信从无人看好到炙手可热、在8个月里横扫市场的背后，是其功能的演进、创新，是“不仅仅是一款聊天工具”理念的胜利，是鼎鼎有名的Foxmail团队、是张小龙式的胜利。



吴刚

入选理由：从传统游戏转型至移动游戏，并获得巨大成功的程序员代表，国内移动应用领军人物。

在盈利困难的移动互联网时局下，其代表作《二战风云》凭借最高6万美元的日收入，成为业界标杆。《二战风云》成功以来，模仿它的游戏就有几十款。虽然游戏内容可以被抄袭，但作为游戏界老兵，吴刚对用户心理的拿捏，对市场的理解，对游戏世界观的构架，对用户品质的控制，对游戏运营掌控，对团队发展的控制，都是一般开发者难以望及项背的。与其盯着《二战风云》不放，不如看看吴刚做了哪些事儿：稳扎稳打保障游戏高品质，保持团队稳定性，大力培养人才。



### 张宏江

**入选理由：**放弃全球一流外企微软亚洲工程院院长的头衔，转投民营企业金山。

2011年10月24日，身为金山董事长和风险投资者的雷军特地穿上西装，欢迎前微软亚洲工程院院长张宏江出任金山上市公司金山软件CEO。张宏江坦诚，离开微软和13年前离开硅谷的原因一样，促使自己离开的不光是微软，而是当时的经济 and 产业形势。他花了两个月时间考虑自己的职业规划，最终金山本身的历史和目前的产业大势让自己下定决心加盟。尽管还没有具体的规划，但张宏江认为，移动互联网是互联网的一大趋势，也将是金山未来的主要投入增长点。张宏江的加盟是否会让金山开启新局面，令人期待。



### 云风

**入选理由：**程序员创业大潮的代表。

2011年9月，网易杭州研究中心总监云风离职消息引发热议。1999年初开始制作二维游戏引擎——风魂系列，被多家公司和小组用于游戏制作。他是网易主力游戏《大话西游》、《梦幻西游》的主力开发者。这两款游戏颇受玩家喜爱，创造了国产网络游戏的高潮。云风在离职信中透露出对技术发展的追求，希望在新项目中有所突破。不久，他证实自己已到广州与前同事成立简悦公司，还是做游戏。



### 周娟

**入选理由：**创立的56网被人人网高价收购，国内最成功的女程序员。

人人网宣布将以8000万美元全资收购56网，关注的目光又一次聚集在56创始人周娟身上。从1999年进入网易算起，周娟算是个不折不扣的互联网老兵了。

2005年5月，周娟从网易辞职，一头扎进了56的视频天地，享受工作和生活结合在一起的乐趣。56的团队一直保持罕有的精干，业务却一直在飞速的发展。2011年9月27日，人人网宣布全资收购56网，周娟作为56网创始人、总裁兼CEO，将出任人人网副总裁。巾帼不让须眉，从网易、56到人人网，这位国内最杰出女程序员的传奇还在延续。



### 林仕鼎

**入选理由：**云计算技术派。百度云计算总工程师，首次提出数据中心计算概念。

林仕鼎在中国计算机大会提出，比起云计算，数据中心计算在技术上是更合适的表述。总结来说：1. 单用户多任务到多用户单任务的环境变化，导致架构师对latency vs thought、fairness vs efficiency有了新的审视；2. 架构由层次化向竖井式演进，系统由需求驱动而定制；3. 由于故障引入和更多的数据处理应用，确定性系统转变为不确定系统，设计理念和评价指标都很不同。在数据中心环境中，通常有完整设计的存储系统，优化过数据组织和读写，对存储设备的需求就是基本的定长block。



### 吴军

**入选理由：**《浪潮之巅》清晰地分析了科技产业、商业机遇内在规律，引发了IT人员对IT产业的深层思考。

吴军的《浪潮之巅》一书以一波波的“浪潮”为主线，把离散的IT公司连成一个整体，清晰地辨析了时代浪潮的准则，解读了公司基因的力量，盘点了无数杰出企业和领袖。并在此基础上，对未来的发展做出推论。因为对世界IT产业高屋建瓴的视角，深入浅出的剖析，而获得盛誉。《浪潮之巅》无疑是一本帮助IT人理解自身在产业内位置坐标和未来走向的指南，除了其内容带来的深远影响之外，身为腾讯副总裁的吴军作为一位具有的广阔视野、渊博学识的科学家也为大家树立了榜样。



### 季逸超

**入选理由：**Macworld 2011特等奖获得者，国内新生代代表人物。

从小受父亲影响，学习计算机技术。从零元起步，在为中关村某企业写商业计划书掘到第一桶金后开始不断开发不同类型的软件，在某浏览器公司拿70元的日薪，自己开发《猛犸浏览器》……季逸超同学的成长轨迹令人钦佩，但《猛犸浏览器》成功之下，其清晰的思路更值得挖掘：一个人的团队如何克服技术、美工、推广、本地化、运营障碍，始终保持团队独立、思想独立。虽然《猛犸浏览器》算不上一款伟大的产品，但其代表的是移动时代小型团队的成功，想必才上大一的季逸超，会在未来带给我们更大的惊喜。



# 年度十大产品（国外篇）

## TOP 10榜单

- Android 4.0
- Chrome OS
- iPad 2
- Windows Phone 7
- Siri
- Linux Mint
- Google+
- Watson
- iCloud
- Kindle Fire



**Android 4.0**  
出品方：Google

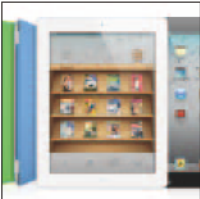
Android在全球智能手机操作系统的市场份额已达48%，是最受欢迎的智能手机平台。最新版Android 4.0代号为冰淇淋三明治

（Ice Cream Sandwich），有包括主题、模式、用户界面等方面的改进。在此重大更新中，Google将实现四大改进，以期在同一高度与对手竞速：一、新一代Android系统将只有一个版本，智能手机和平板电脑融合，打通产品线，提升运行速度；二、内置Google服务Google+和Google Music等，提供音频和视频的Android Market资源下载，并提供类似Game Center的游戏中心；三、全新设计的用户体验界面；四、集成Google TV和Chrome OS的智能停放。



**Chrome OS**  
出品方：Google

Chrome OS是按照Google极简设计理念完成的操作系统，很像Google Chrome浏览器。Google认为在未来大部分用户界面和应用程序将从本地桌面的环境迁移至互联网上。云计算是这一图景中至关重要的一部分。无论软件还是硬件工程师，Google都以“网络即平台”的观点鼓舞他们投入这一图景的建设中来。对于如火如荼的Android系统发展，Google明确指出，前者是专为移动设备创造的。而Chrome OS是设计为大部分时间都花在互联网的用户使用的，将运行在上网本和台式电脑上。Chrome OS只与Google自造的硬件捆绑销售，限制了它的普及，但这是Google创造未来的一个很好的行动。



**iPad 2**  
出品方：Apple

iPad 2是Apple公司推出的第二代iPad产品。与一代相比，最大的区别在于iPad 2更为轻薄并且拥有前置摄像头，并且同时发售黑白两种颜色的版本。除了更轻、更薄，iPad 2承袭了Apple公司各款产品的特点，顶尖的用户体验设计，具备优秀的娱乐影音功能。iPad 2完全奠定了Apple在平板电脑领域的标杆地位与事实标准，也使平板电脑这一种类的产品更加深入人心。



**Windows Phone 7**  
出品方：微软

Windows Phone 7前身为Windows Mobile，但它完全变革了人们对原本系统的印象，将微软旗下的Xbox LIVE、Zune、MSN和Bing等产品整合其中，采用Metro作为界面语言，并严格控制它的硬件规范。此版本不再以企业市场为目标，而是更关注一般的消费者市场。微软CEO鲍尔默评价它为“契合用户生活脚步、满足社交需求并提供各种无缝体验。”正是从此版本开始，微软有了与iOS、Android平台媲美的产品，可认为是其移动战略的转折点。此外，与Nokia的深度研发合作，也帮助微软迅速建立起品牌生态系统。





### Siri

出品方: Apple

Siri不是简单的语音识别或者语音指令,这些都是非常简单的语音模式匹配,而Siri是真正的人工智能,或者说这是第一次人工智能技术成熟到可以进入消费领域被普通人使用。Siri提供了一种新的交互方式,想想当年的多点触摸技术出现,同样是提供了一种新的交互方式,在之后的几年这种交互是如何改变了用户使用设备和应用的方式。对于Siri也是如此。目前Siri的成熟度还不够,还没有提供API给普通开发者,但相信这只是时间问题。从提供数据源到完成终端应用,围绕这种新技术可做的事情非常多,现在已经可以开始想象未来了。



### Google+

出品方: Google

Google+是Google推出的社交网络服务,几乎整合了Google全线产品。在与其他社交服务的区别方面,Google提出了更好的隐私管理方式,其要点是朋友和熟人的Circles,按照其产品概念,用户可以根据不同圈子组织联系人、分成群,根据需要分享照片、视频以及其他信息,让分享最优化。此外,Google+还支持Hangouts的多人视频聊天功能。Google+还与Android系统集成,具有更好的移动特性,并结合LBS,以达成LBS与SNS的深度结合。作为Google在社交领域的又一次努力,Google+或许让人们在Facebook和Twitter之外,有一个可以适当分享的地方,如果成真,那将是社交网络的另一场革命。



### iCloud

出品方: Apple

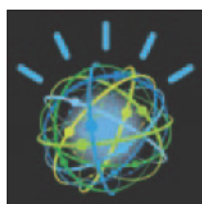
iCloud是Apple公司提供的云端服务,功能是存储内容,包括将购买的音乐、应用和电子书等推送到所有设备,目前提供给用户5GB的免费存储空间。2011年5月31日,Apple官方首次宣称有iCloud,并将取代不太成功的MobileMe,它是基于原有的MobileMe功能全新改写而成,提供了原有的邮件、日历和联络人同步功能。2011年6月6日,Steve Jobs抱病主持WWDC大会,正式发布iCloud。与Google和Amazon的云端音乐服务相比,iCloud具有独特的“扫描配对”功能,并提供包括办公生产率、开发工具、媒体和窗口小部件等在内的应用程序。



### Linux Mint

出品方: Linux Mint社区

Linux Mint是一款年轻的Linux发行版,始于2006年。它基于相当成熟的软件管理,并采用了最新的Linux内核。它基于Ubuntu和Debian项目,但是获得了比两者都要高的流行度。究其原因Linux Mint强大的可安装软件库和非常完善的服务设置机制。Linux Mint遵循每六个月发布新版本的制度,保证用户及时用上新技术。Linux Mint获得广泛关注的另一个原因也许还在于Ubuntu采用新的Unity图形界面,这导致很多原有用户流失,转向更为稳健成熟的Linux Mint。



### Watson

出品方: IBM

Watson是IBM最新的电脑系统,最初得名于IBM创始人Thomas J. Watson。开发它的目的不是为了参加比赛,而是为了了解词汇、语言和人类知识中的更复杂领域。为摹仿人脑,Watson装载了大量文档。2011年2月14日,史上最强的人机对抗拉开序幕。Watson参加了美国智力节目Jeopardy,挑战两位人类选手Ken和Brad,结果Watson获胜。IBM目前正在尝试Watson的商业化,比如将Watson的技术在医疗行业内实现商业应用。然而,要使Watson真正成为人类的得力助手还需要改进。



### Kindle Fire

出品方: Amazon

如果只谈一台平板电脑,那么除了价格便宜,Kindle Fire实在没什么特点。但如果考虑到Amazon的运营模式,从Fire可以看出其背后庞大的野心,那就非常值得一谈。Fire价格便宜的原因是配置低,而配置低的平板电脑为什么也能上市,是因为Amazon用了他们称为Silk的技术,在云端渲染内容,降低了终端设备的计算负担。就目前使用情况看,Silk并没有如之前预想的那么快,应该还需要更多优化。不过在所有厂商竞争终端设备配置的时代,这样做显得很特别,用户确实可以以更便宜的价格获得一台足够使用的平板电脑。

# 年度十大产品（国内篇）

## TOP 10榜单

小米手机    天河一号    微信    豌豆荚    二战风云    网易新闻客户端  
海豚浏览器    知乎    淘宝OceanBase    迅雷云点播



### 小米手机

出品方：小米公司

小米手机是小米公司推出的一款智能手机，2011年8月发布。1999元的售价被认为在智能手机中有相当的竞争力，宣传重点针对手机发烧友，采用线上销售模式，生产由英华达代工，操作系统采用小米公司基于Android深入定制的MIUI操作系统。小米手机已于2011年11月正式上市，销量良好。

小米手机发布伊始就受到舆论的广泛关注，虽然上市不到20天就爆出“质量门”事件，但随后官方做了妥善回应。小米手机获得众多的关注反映了国内移动互联网的热度不减。



### 天河一号

出品方：国防科学技术大学

“天河一号”由国防科学技术大学研制，部署在国家超级计算机中心，运算速度可以达到每秒2570万亿次。作为我国首台千万亿次超级计算机，“天河一号”从2010年9月开始进行系统调试与测试可在资源勘探、生物医药研究、航空航天装备研制、金融工程、新材料开发等方面得到广泛应用。

2011年，国际TOP500组织在网站上公布了最新全球超级计算机前500强排行榜，“天河一号”名列第二。在国际上的抢眼表现，我国在高性能计算机方面的投入可见一斑。



### 微信

出品方：腾讯

微信是腾讯推出的一款为智能手机提供即时通讯服务的免费应用程序。微信支持跨通信运营商、跨操作系统平台通过网络快速发送免费语音短信、视频、图片和文字，支持多人群聊的手机聊天软件。目前微信支持iOS、Android和Symbian平台。

腾讯对手机语音聊天工具的投入表明了对此领域产品布局的重视，也说明此领域是未来的移动互联网的发展热点。



### 豌豆荚手机精灵

出品方：豌豆实验室

豌豆荚手机精灵是创新工场首个曝光投资项目，Android手机管理软件。连接手机和电脑后，可通过豌豆荚手机精灵在电脑上管理手机中的通讯录、短信、应用程序和音乐等，也能在电脑上备份手机中的资料。

手机同步管理软件产品众多，豌豆荚能够得到广泛的用户，其产品团队在用户体验和产品细节方面下的工夫值得肯定。



## 二战风云

出品方：顽石互动

《二战风云》是顽石互动推出的一款多人在线的战争策略游戏。玩家需要对城市进行管理，建立生产、科研和贸易等体系，采用军事或外交手段与其他玩家交互。游戏主要面向35岁以下的年轻人群。

创作团队对原本晦涩难懂的战争设置做了便于理解和记忆的简化，打造成了一款易被大众接受的战争游戏产品。



## 网易新闻客户端

出品方：网易

网易新闻客户端是网易发布的移动端新闻阅读软件。为手机用户提供全天24小时滚动即时的网易新闻资讯，此外还具有突发新闻推送功能，同时，用户还能由此参与具有网易新闻特色的跟帖盖楼，与网站保持同步。

作为传统的门户网站，网易在新型终端上取得了不俗的成绩，这是新媒体时代勇于变革所收到的最好褒奖。



## 海豚浏览器

出品方：MoboTap

海豚浏览器是由MoboTap开发制作的智能手机浏览器。该款浏览器基于WebKit内核，并有超过50种可扩展的插件以及个性化定制功能。海豚浏览器一大亮点就是手势操作功能，用户可以自定义专属手势，来完成绝大部分操作。同时，海豚浏览器还在Android平台上支持多点触控和手势操作。

海豚浏览器以优良的用户体验见长，在触摸屏幕上力求能获得比键盘输入还要快捷的人机交互。此外，在社交化等领域的积极开拓，也展现了其不混同他人的产品思路。

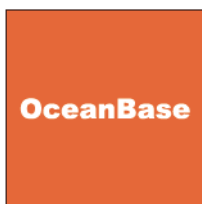


## 知乎

出品方：知乎团队

知乎是由知乎团队提供的一家“社会化问答网站”，2011年1月26日上线，采用邀请制公测，被称为中国的Quora，目前已得到创新工场的天使投资。知乎试图建立起社会网络，期望以关系社区形式来帮助用户找到更好的问题和答案。在获得的问题答案中，采用了类似 Digg 的支持机制。

与Quora相同，知乎也面临着初始用户热情下降、答案质量下降的问题，但是作为国内问答网站的一次有益尝试，知乎值得肯定。



## 淘宝OceanBase

出品方：淘宝

淘宝的商品、交易、订单和购物爱好等数据随着业务的快速发展，数据量也急剧膨胀，记录数从几千万条增加到数十亿条，数据量从百GB增加到数TB，传统的关系型数据库已经无法承担如此海量的数据。

OceanBase是淘宝技术团队解决不断增加的结构化数据存储与查询等问题的最新努力。



## 迅雷云点播

出品方：迅雷公司

云点播是迅雷公司推出的在线视频服务。用户只需提供BT种子或者下载地址就可以直接在线观看。其技术核心是云转码，从而达到比普通边下边播，响应更迅速、播放更流畅的效果。

云点播是迅雷公司以自身技术优势，结合本土视频服务特点做出的一次创新，一经推出，就受到了大量用户喜爱。



# 年度十大技术

文 / 杨海朝 杨栋

## TOP 10榜单

DevOps	Git	Hadoop	HTML5	jQuery Mobile
Node.js	NoSQL	Puppet	RESTful	响应式Web设计

## DevOps

在速度就是市场的时代，产品需要快速开发和发布，人们开始关注DevOps的思想，希望能对产品的交付、测试以及功能开发和维护工作产生积极的影响。当前很多开发团队与运营团队之间存在一定的“鸿沟”，运维团队要求更好的可靠性或变更很少，开发团队则希望发布和变更更快，而业务团队的需求则是更快地将新功能发布出去给用户使用。如何去掉这个鸿沟，让开发团队与运维团队之间具有高效的协作关系，就需要DevOps思想的引入。它能改善团队间协作关系，提升整个组织的工作效率，降低频繁变更带来的生产环境的风险。

## Git

应用Linux内核开发的版本控制工具Git，与常用的版本控制工具Subversion、CVS等不同，它采用了分布式版本库的方式，不必服务器端软件支持，使源代码的发布和交流极其方便。Git速度很快，具有超强的合并跟踪能力，没有对版本库的浏览和修改做任何的权限限制。

尽管最初Git的开发是为了辅助Linux内核开发的过程，但现在越来越多的开源项目采用Git来管理项目开发。

## Hadoop

2011是Hadoop普及年。MapR、Hortonworks、EMC、Cloudera等公司陆续发布Hadoop产品，微软、IBM、Dell、NetApp等公司也开怀拥抱Hadoop，除了百度、淘宝、腾讯，国内其他互联网公司也开始在内部大规模使用Hadoop系统。

2011又是Hadoop进步年。YARN和MapReduce 2.0的诞生推动了Hadoop的技术发展，Mesos也成为开源项目，还有许多新兴的并行编程系统逐渐流行起来，像Spark、Giraph和Accumulo等。

2011更是Hadoop感染年。Hadoop作为一个完整的云计算生态系统，孕育了大批热爱技术、热爱生活的程序员，是Hadoop将全球的程序员联系起来，大家跨越公司、跨越国家，实现了真正意义上的技术共享、技术交融。

2012将又是一个Hadoop年，它将成为计算机领域的图腾……

## HTML5

借助HTML5，开发者可以轻松创建更丰富、互动性更强的应用。而随着开发者疲于为Android、iOS和WindowsPhone等各类移动操作系统开发应用，HTML5将使得开发者“一次开发，随处可用”成为可能。

此外，Adobe取消移动版Flash，宣告了Flash这种重要的移动视频和应用开发工具将会消失。显然，这样的利好将加速HTML5的崛起。同样，对于消费者而言，HTML5将使他们拥有更为丰富的应用和体验。

但就像所有伟大的技术成长之路一样，HTML5目前还存在标准不够完善、技术不够成熟的问题，不过我们相信，HTML5的爆发指日可待。

## jQuery Mobile

jQuery目前已经俨然成为JavaScript事实上的代名词，这得益于其所具备的优秀的可扩展性、易用性、可控性。而随着与平板

电脑和智能手机等设备应用相关的触屏优化技术和Web框架的发展, jQuery Mobile又成为了跨平台、支持多设备的手机应用的非常重要的技术选择之一。

越来越多的开发者在使用HTML5来开发移动应用, 建立一个稳定的易用性好的JavaScript框架将成为移动技术开发人员的重要选择。jQuery Mobile具备成为这一框架的事实上的基础。相信2012年的jQuery Mobile会与HTML5一起有更大的突破。

## Node.js

采用Google V8 JavaScript引擎的Node.js上手非常容易, 它通过更改连接到服务器的方式, 每个连接发射一个在Node引擎的进程中运行的事件, 而不是为每个连接生成一个新的OS线程(并为其分配一些配套内存), 能支持数万个并发连接, 性能很好。目前Node社区非常火, 具有上千个开源的Node.js模块。Node.js在WebFramework、WebSocket、RPC、模板引擎和数据抓取服务等领域具有很多应用。Node.js使用无阻塞模式和事件驱动模型, 这个模型非常高效、可扩展性非常强, 同时它的实现代码小且易于阅读, 使编写一个高度可伸缩的解决方案变得比较容易。Node.js在Web开发人员中非常受关注, 它允许开发人员同时在客户端和服务端上编写JavaScript。未来会有越来越多的应用使用Node.js来实现。

## NoSQL

随着越来越庞大的互联网产品的出现, 例如微博、SNS应用等, 对关系数据库在可扩展性和高并发方面提出了新的挑战。NoSQL的出现主要是为了解决传统关系型数据库不能解决或不能很好解决的应用场景, 并且要求进行分布式部署。这一年各种NoSQL产品发展很快, 例如Redis、MongoDB。很多互联网公司在不断地做着各种尝试, 把它放到合适的位置去解决自己应用中遇到的问题。NoSQL是not only use SQL的缩写, 是关系型数据库的有力补充, 但NoSQL也不是银弹, 索引数据还需要用关系型数据库进行存储, 内容类数据可以逐步转向K-V存储。

## Puppet

基于Ruby的开源系统配置管理工具, 通过使用自身的Puppet描述语言, 简化资源的管理和处理这些资源间的各种关系, 同时

让系统管理员可以通过这个强大的框架, 相互交流和共享成熟的工具, 避免重复的劳动, 使资源的管理工作更快地完成。特别是在服务器规模处于万级时, 让它的所有配置工作整齐统一。

Puppet具有三个重要的特性: 一、它的语法允许你创建一个单独脚本, 用来在你所有的目标主机上建立一个用户, 所有的目标主机依次使用适用于本地系统的语法解释和执行这个模块; 二、具有非常好的灵活性; 三、非常容易扩展。目前Puppet还处于开发和发展中, 它的社区也在快速壮大, 有很多好的思想在不断地融入, 社区开发者和捐献者还在不断增强Puppet的功能。

## RESTful

RESTful作为目前一种互联网软件架构, 其结构清晰、扩展方便并且易于理解, 被越来越多网站所采用。RESTful的几个约束条件作为一个整体应用时, 将生成一个简单有效、可扩展、安全性好的架构。由于它的轻量级以及可以直接通过HTTP传输数据的特性, RESTful Web服务逐步成为基于SOAP服务的一种替代方案。在动态Web服务的多层架构可以实现可重用性和组件可响应性的清晰隔离。

RESTful的架构风格是一种全新Web应用的开发风格, 随着RESTful架构成为主流技术, 为人们提供了一种全新的互联网网络应用开发的思维模式: 通过URL来设计系统结构。很多架构设计会从SOAP逐渐淡出, 越来越向RESTful思想靠拢。

## 响应式Web设计

硬件设备特别是分辨率的发展, 使得Web设计在被慢慢落下。为每种新设备及分辨率创建独立的版本似乎并不实际。

这时, 响应式Web (Responsive Web Design) 设计横空出世。其理念是页面的设计与开发应当根据用户行为以及设备环境(系统平台、屏幕尺寸、屏幕定向等)进行相应的响应和调整, 包括弹性格和布局、图片、CSS等。因此, 无论笔记本还是iPad, 页面都应该能够自动切换分辨率、图片尺寸及相关脚本功能等。这样, 我们就可以不必再为新设备做专门的版本设计和开发了。可以说, 响应式Web设计这种全新的设计思维模式必将引领设计领域的一场变革。

# 年度十大图书（外版）

## TOP 10榜单

《设计原本》      《项目百态》      《黑客与画家》      《MongoDB权威指南》      《启示录》  
《智能Web算法》      《REST实战》      《持续交付》      《触动人心》      《史蒂夫·乔布斯传》



### 设计原本

作者：Frederick P. Brooks, Jr.  
译者：InfoQ中文站、王海鹏、高博  
出版社：机械工业出版社

设计是一个通过反馈来学习的过程。设计师在开始设计时，并非什么都懂。需要与很多人交谈，画出许多模型草稿，一次次否定前面的想法，用相当长的一段时间切磋琢磨。设计师在这个过程中学到很多：设计要经过实际环境的考验；设计是造物主赋予我们的一项特殊才能；设计是一种乐趣；知之者不如好之者，好之者不如乐之者。你可能不是软硬件设计师，但只要你喜欢设计，就会喜欢Brooks在《设计原本》中讲述的设计故事。



### 项目百态

作者：Tom Demarco、Peter Hruschka、Tim Lister、Suzanne Robertson、Steve McMenamin、James Robertson  
译者：金明  
出版社：人民邮电出版社

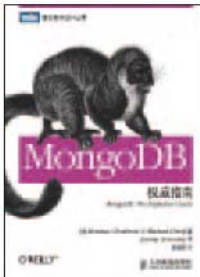
作者从实际经验中，提炼出86种项目行为的模式，并为每种模式取了直观的名字、配了关联图片和点题说明。书中以有趣的故事引出每个模式，然后分析问题的根源和特征，最后做出意味深长的总结。这样的写法，让读者在阅读中既不时悚然于问题的似曾相识，又常常拍案叫绝：原来软件项目可以这样玩！在软件行业影响深远的《人件》的两位作者，20年后再次携手完成的这部第19届Jolt大奖作品。



### 黑客与画家

作者：Paul Graham  
译者：阮一峰  
出版社：人民邮电出版社

这是“硅谷创业之父”Paul Graham的文集，也是一名顶级黑客以自身的视角给我们解读计算机的世界和时代。黑客拥有基于计算机的特殊手艺，黑客拥有创作的天赋和热情。黑客的叛逆天性，主导了他们看待世界的角度，左右了他们使用技术的方式。对大众而言，阅读这本书会沉浸于黑客的奇妙世界而不能自拔；对黑客而言，阅读这本书将叹服于顶级黑客的洞察力和概括性，并被迫重新思考计算机编程的本质，甚至转而拾起古老的LISP。

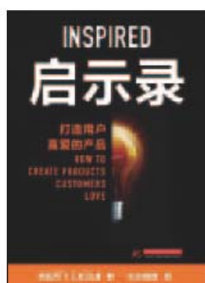


### MongoDB权威指南

作者：Kristina Chodorow、Michael Dirolf  
译者：程显峰  
出版社：人民邮电出版社

MongoDB如何管理通过Web应用收集的海量数据呢？通过本书，你会了解面向文档数据库的诸多优点，会发现MongoDB如此稳定、性能优越甚至能够无限水平扩展背后的原因。本书作者来自开发MongoDB的公司10gen。数据库开发人员可将此书作为参考指南，系统管理员可以从本书中找到高级配置技巧，其他用户可以了解一些基本概念和用例。你会发现，将数据组织成自包含的JSON风格的文档比组织成关系型数据库中的记录要容易得多。





## 启示录

作者: Marty Cagan

译者: 七印部落

出版社: 华中科技大学出版社

怎样发掘有价值的产品? 拿什么说服开发团队接受你的产品设计? 如何将敏捷方法融入产品开发? 过去20多年, Marty Cagan作为高级产品经理人为多家一流企业工作过。他亲历了个人电脑、互联网、电子商务的起落沉浮。在本书中Marty Cagan总结20余年软件产品的管理经验,直指产品研发流程存在的弊病,为产品经理答疑解惑、指点迷津。从人员、流程、产品三个角度介绍了现代软件(互联网)产品管理的实践经验和理念。



## REST实战

作者: Jim Webber, Savas Parastatidis, Ian Robinson

译者: 李锐、俞黎敏、马钧、崔毅

出版社: 东南大学出版社

在这本富有洞察力的书中,三位SOA专家对于REST进行了讲求实际的解释,并且通过将Web的指导原理应用到普通的企业计算问题中,向你展示了如何开发简单、优雅分布式超媒体系统。你将会学习到很多技术,并且随着一家典型的公司从最初的小企业逐渐成长为全球化的企业,使用这些Web技术和模式来解决这家公司在成长过程中产生的各种需求。



## 触动人心

作者: Josh Clark

译者: 包季真

出版社: 电子工业出版社

可能你设计网站产品或软件界面早已得心应手,可是遇到了iPhone,却感觉无从下手。无论你是产品经理、设计师、创业者还是程序员,本书都能告诉你如何从iPhone的角度来思考应用设计。本书能帮助你理解如何设计iPhone应用,要创建一款触动人心的应用,需要如何去综合思考设计、心理、文化、人体工程和可用性问题。在书中,你会看到大量的真实应用的设计实例和思考过程。



## 智能Web算法

作者: Haralambos Marmanis, Dmitry Babenko

译者: 阿稳、陈钢

出版社: 电子工业出版社

本书涵盖了五类重要的智能算法: 搜索、推荐、聚类、分类和分类器组合,并结合具体的案例讨论了它们在Web应用中的角色及要注意的问题。除了第1章的概要性介绍以及第7章对所有技术的整合应用外,第2~6章以代码示例的形式分别对这五类算法进行了介绍。本书中的例子和思想应用广泛,所以对于希望从业务角度更好地理解有关技术的技术经理、产品经理和管理层来说,本书也有一定的价值。



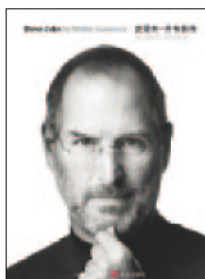
## 持续交付

作者: Jez Humble, David Farley

译者: 乔梁

出版社: 人民邮电出版社

本书不单提出了概念,还提供一整套的原则、方法和实践,按图索骥即可持续交付。无论是产品经理、项目经理、开发人员,还是测试人员、运维人员,甚至公司高层、销售人员,都可以从这本书获益。Dr. Dobbs杂志为它授予2011年Jolt大奖时,评论道:“已经很长时间没有出现这种能够颠覆产业规则的图书了”,并说:“重新定义了敏捷过程和持续集成。其影响力之于过程领域,将不啻于1999年Martin Fowler的《重构》之于代码领域。”



## 史蒂夫·乔布斯传

作者: Walter Isaacson

译者: 管延圻、魏群、余倩、赵萌萌

出版社: 中信出版社

史蒂夫·乔布斯本人表示,《史蒂夫·乔布斯传》涉及他曾经犯过的错误、失败的经历以及私生活,有些事情是他自认为并不得意的,但没有什么是不可公诸于众的。史蒂夫·乔布斯的激情、精力、欲望、完美主义、艺术修养、残暴还有对掌控权的迷恋塑造出的商业哲学一览无余。同样,史蒂夫·乔布斯的朋友、敌人,还有同事在《史蒂夫·乔布斯传》里也为我们提供了一个前所未有的毫无掩饰的视角。

# 年度十大图书（本版）

## TOP 10榜单

《结网》      《Oday安全：软件漏洞分析技术》      《Android技术内幕：系统卷》      《Git权威指南》  
《浪潮之巅》      《暗时间》      《你必须知道的.NET》      《构建高可用Linux服务器》  
《C++ 反汇编与逆向分析技术机密》      《专业嵌入式软件开发——全面走向高质高效编程》



**结网**  
作者：王坚  
出版社：人民邮电出版社

本书以如何创建、发布、推广互联网产品为主线，介绍了互联网产品经理的工作内容以及应对每一部分工作所需的方法和工具。为用户创造价值是产品经理的第一要务，产品经理的工作是围绕用户及具体任务展开的，本书丰富的案例和透彻的分析道出了从发现用户到最终满足用户这一过程背后的玄机。新版完善了各章节，增加了优雅降级等内容，读者也可从中更深地去感受一名产品经理的感悟。



**Oday安全：软件漏洞分析技术（第2版）**  
作者：王清、张东辉、周浩、王继刚、赵双  
出版社：电子工业出版社

Oday指在软件开发商知晓并发布相关补丁前已被某些软件分析者掌握的可用于攻击的漏洞。对于信息安全相关人员，Oday漏洞是最严重的安全威胁，而对于攻击者而言，Oday漏洞就是他们的秘密武器。由于相关技术资料的敏感性，国内可见的相关研究和分析资料非常稀少。本书系统地介绍了Windows平台软件漏洞分析的原理与技术细节，结构清晰，内容组织充分考虑了国内读者习惯，非常适合国内的IT安全相关技术人员学习和参考，是一本难得的好书。



**Android技术内幕：系统卷**  
作者：杨丰盛  
出版社：机械工业出版社

《Android技术内幕》系列图书分为两卷：系统卷和应用卷，完全立足于Android系统的源代码，深度解析Android内核的架构设计与实现原理。系统卷透彻分析了Android内核层、硬件抽象层和系统运行库层的机制和实现原理，在宏观上能帮助读者厘清Android系统架构各层之间的复杂关系和Android系统的架构设计，在微观上能帮助读者深入地了解Android系统中每个功能模块的实现细节，从而帮助开发者解决他们在进行系统开发时可能会遇到的各种难题。



**Git权威指南**  
作者：蒋鑫  
出版社：机械工业出版社

本书是Git领域的集大成之作，同时也是版本控制领域的经典著作之一，广度、深度和实战性兼备。从基本操作和使用、团队协作、设计原理、使用模式、版本库迁移和服务器架设等数十个方面对Git进行了全方位的讲解，将Git的功能、特性、使用方法和原理等阐述得淋漓尽致，能解答读者在实际学习和使用中遇到的很多问题。本书得到了Git官方维护者Junio C Hamano等数位专家的高度认可和极力推荐。出版后在社区内引起了强烈反响，广大读者给予了极高的评价。





## 浪潮之巅

作者：吴军  
出版社：电子工业出版社

近一百多年来，总有一些公司很幸运地、有意识或无意识地站在技术革命的浪尖之上。在这十几年间，它们代表着科技的浪潮，直到下一波浪潮的来临。从一百年前算起，AT&T、IBM、苹果、英特尔、微软、思科、雅虎和Google都先后被幸运地推到了浪尖。虽然，它们来自不同的领域，中间有些已经衰落或正在衰落，但是它们都极度辉煌过。本书不仅讲述科技工业的历史，更重在揭示它的规律性。



## 暗时间

作者：刘未鹏  
出版社：电子工业出版社

“我们每个人的生命就像一个个沙漏，每个人的沙漏里装的沙子的总量是相当的——大家都活的差不多长。不同的是，有些人的沙漏颈部较细，有些人的沙漏颈部较粗。那些颈部较细的沙漏能够抓住每一粒时间之沙，虽然沙子总量一样，但却相对拥有更长的生命。”作者在书中如此说。善于利用思维时间的人，可以无形中比别人多出很多时间，从而实际意义上能比别人多活很多年，这也许就是书名《暗时间》的来源。



## 你必须知道的.NET (第2版)

作者：王涛  
出版社：电子工业出版社

这是一本修炼.NET内功的书。通过一本.NET入门书几天就学会开发一些小程序，并根据自己的爱好学些编程技术和技巧，但如果真的想成为.NET的专业高手，想靠它吃饭，靠它发展自己的事业，那么请阅读本书吧。本书没有以往国内书籍抄袭或拼凑文字的浮躁，也没有国外资料因翻译或文化差异所造成阅读的困惑，而更多的是对.NET深层实现的剖析。或许阅读之前，会觉得自己知道的.NET已经很多，但读完本书，会感受到，原来.NET还有很多必须知道的内容并不知道。



## 构建高可用Linux服务器

作者：余洪春  
出版社：机械工业出版社

本书的内容完全来自于实际生产环境，从Linux服务器的虚拟化、生产环境下服务器的故障诊断与排除、生产环境下的Shell脚本、高可用Linux集群建设、VPN在企业中的部署应用、Linux防火墙等多个方面阐述了构建高可用Linux服务器的方法与最佳实践。实践性非常强，包含大量企业级的应用案例及相应的解决方案，读者可以直接用这些方案解决在实际工作中遇到的问题。本书得到了多个运维社区和来自腾讯等多家著名IT公司运维专家的强烈推荐。



## C++ 反汇编与逆向分析技术揭秘

作者：钱林松  
出版社：机械工业出版社

本书出版在安全类图书领域中有重要的里程碑意义。理论与实践并重，理论部分系统地讲解了C++的各种语法特性和元素的逆向分析方法和流程，重在授人以渔；实践部分通过几个经典的案例演示了逆向分析技术的具体实施步骤和方法。作者钱林松是逆向分析领域的资深专家，从事计算机安全和软件开发工作12年，对C/C++技术和Windows的底层机制也有非常深入的研究。本书因为内容翔实、讲解深入而深受读者欢迎，首印不到半月全部售罄。



## 专业嵌入式软件开发——全面走向高质高效编程

作者：李云  
出版社：电子工业出版社

本书覆盖了专业嵌入式软件开发所需掌握的各方面知识，在广度和深度上达到了很好的平衡。对于一个初学者，要在较短的时间掌握嵌入式软件开发所需的理论知识和实践技巧是很具挑战的任务。理论知识可以通过广泛阅读来获得，实践技巧也可在工作中慢慢积累，但这些方法都是费力和低效的。作者基于自己十余年的软件开发经验，将精选的知识、方法和思想进行浓缩，帮助初学者以较轻松的方式越过第一道门槛。



# 2012年十大期待

## 1 云计算落地

云计算从正式提出至今，吵吵嚷嚷也有好几年了。尽管期间也有一些鲜活的案例，昭示了云计算的强大威力，但大多数人要么“云里雾里”，要么“满腹疑云”。云计算普及的障碍很多，除了安全性问题之外，还有应用缺乏等。期待云计算落地在2012年。

## 2 Windows 8

微软在2011 BUILD大会上隆重推出的基于触摸操作“Metro”界面的Windows 8，已经给了业界内外极大的视觉震撼。作为微软史上自Windows 95后重新构建的第二款重大操作系统，其推出后的实际表现究竟如何，2012年立见分晓。

## 3 更好的互联网

互联网已成为人类另一个栖居地，我们建设出了一个繁茂的虚拟世界。历史上从未有如此的盛况，所有的知识等待着互通有无，所有的想法等待着探求与分享。我们也从未对一种事物抱有这么大的期许与信心：相信互联网能够填平沟壑、冲破阻碍。我们期待：信息之传达无远弗届，技术之进步福泽大众。

## 4 开源硬件

鼓捣开源硬件往往被看成是死宅或Geek的行为。为了破除这种偏见，我们主张开源硬件要如同自由及开源软件，并由它们所驱动。无论电路图还是材料清单，都要自由开放其设计。我们期望开源硬件的材料成本更低、更易获得；设计硬件的软件更加友好、技术门槛更低。普通人也能按自己的愿望设计出个性化的机器人、音箱、电脑、打印机甚至家具。

## 5 Windows Phone成为第三极

苹果在乔布斯离世之后的创新力在一定程度上让人有所怀疑，加之Android阵营的貌合神离、RIM和webOS的没落，或许能为Windows Phone提供一些机会。当务之急，是要真正打动

更多用户，让他们买回家，满意，并形成口碑效应。未来12个月将是Windows Phone发展的关键阶段。

## 6 IE6市场份额剧降

由于大量盗版XP系统的存在，使得IE6自2001年发布以来，一直统治着国内的浏览器市场。让IE6推出历史舞台，不是一朝一夕的事情，但随着越来越多的网站宣布不再支持IE6以及微软灭掉IE6的决心，使我们完全有理由相信，2012年IE6的市场份额会急剧减少，甚至或许就是IE6的末日。

## 7 BAT格局被打破

近几年来，BAT（百度、阿里、腾讯）三大加上坊间所称雷军系、周鸿祎系、新浪等互联网巨头一直垄断着各自所在的领域，并且它们都在向“全业务”的方向发展，对整个产业的贡献和破坏并存。我们希望看到百家争鸣、百花齐放。

## 8 做好产品的人有福

做一款好的产品，并非易事。而一旦做出来，就会让人们的生活更美好。我们衷心祝福这些做好产品的人都有福气，别的不必说，至少都能有张船票。

## 9 新型汽车

雷诺近日发布了一种搭载Android系统的平板电脑R-link，可通过声音识别或方向牌上的按钮来对它进行操作，是专为汽车打造的一个媒体交互平台。而Google一直在推进的无人驾驶汽车也足足吸引了人们的眼球。这些以前只能在科幻片里面才能见到的新型汽车，或许2012年就飞驰在街头，到时不要太惊讶。

## 10 智能电视

《乔布斯传》中提到乔布斯在去世之前已经搞定了智能电视。他会再次颠覆一大产业吗？如果将广告计入，这又是一个数千亿美元的市场。

# 提高面试代码质量的三要素

文 / 何海涛

作者总结自己多年面试他人以及被他人面试的经验，发现应聘者可以从代码的规范性、完整性和鲁棒性三个方面提高代码的质量。

程序员在职业生涯中难免要接受编程面试。有些程序员由于平时没有养成良好的编程习惯，在面试时写出的代码质量不高，最终遗憾地与心仪的公司和职位失之交臂。因此，如何在面试时能写出高质量的代码，是很多程序员关心的问题。

## 代码的规范性

面试官是根据应聘者写出的代码来决定是否录用一个应聘者的。应聘者首先要把代码写得规范，才可以避免很多低级错误。如果代码写得不够规范，会影响面试官阅读代码的兴致，至少印象分会打折扣。书写、布局和命名都决定着代码的规范性。

■ **规范的代码书写清晰。**绝大部分面试都要求应聘者在白纸或者白板上书写。由于现代人已经习惯了敲键盘打字，手写变得越发不习惯，因此写出来的字潦草难辨。虽然应聘者没有必要为了面试特意去练字，但在面试过程中减慢写字速度、尽量把每个字母写清楚还是很有必要的。不用担心没有时间去写代码。通常编程面试的代码量都不会超过50行，书写不用花多少时间，关键是在写代码之前形成清晰的思路并能把思路用编程语言清楚地书写出来。

■ **规范的代码布局清晰。**平时程序员在集成开发环境如Visual Studio里面写代码，依靠专业工具调整代码的布局，加入合理的缩进并让括号对齐成对呈现。离开这些工具，应聘者就要格外注意布局问题。当循环、判断较多逻辑较复杂时，缩进的层次可能比较多。如果布局不够清晰，缩进也不能体现代码的逻辑，这样的代码将会让人头晕脑胀。

■ **规范的代码命名合理。**很多初学编程的人在写代码时总是习惯用最简单的名字来命名，变量名是i、j、k，函数名是f、g、h。由于这样的名字不能告诉

读者对应的变量或者函数的意义，代码一长就会变得非常晦涩难懂。强烈建议应聘者在写代码时，用完整的英文单词组合命名变量和函数，比如函数需要传入一个二叉树的根结点作为参数，则可以把该参数命名为BinaryTreeNode\* pRoot。不要因为这样会多写几个字母而觉得麻烦。如果一眼能看出变量、函数的用途，应聘者就能避免自己搞混淆而犯一些低级的错误。同时合理的命名也能让面试官一眼就能读懂代码的意图，而不是让他去猜变量到底是数组中的最大值还是最小值。

## 代码的完整性

在面试的过程中，面试官会非常关注应聘者考虑问题是否周全。面试官通过检查代码是否完整来考查应聘者的思维是否全面。通常面试官会检查应聘者的代码是否完成了基本功能、输入边界值是否能得到正确的输出、是否对各种不合规范的非法输入做出了合理的错误处理。

### 三种测试用例确保代码的完整性

应聘者在写代码之前，首先要把可能的输入都想清楚，从而避免在程序中出现各种各样的质量漏洞。也就是说在编码之前要考虑单元测试。如果能够设计全面的单元测试用例并在代码中体现出来，那么写出的代码自然也就是完整正确的了。通常程序员可以从功能测试、边界测试和负面测试三方面设计测试用例，以确保代码的完整性。

■ 首先要考虑的普通功能测试的测试用例。应聘者首先要保证写出的代码能够完成面试官要求的基本功能。比如面试题要求完成的功能是把字符串转换成整数，应聘者就可以考虑输入字符串



“123”来测试自己写的代码。这里要把零、正数（比如123）和负数（比如-123）都考虑进去。

考虑功能测试时，应聘者要尽量突破常规思维的限制，避免忽视某些隐含的功能需求。比如“打印从1到最大的n位数”，很多人觉得很简单。最大的3位数是999、最大的4位数是9999。这些数字很容易就能算出来。但最大的n位数都能用int型表示吗？如果超出int的范围可以考虑long long类型。超出long long能够表示的范围呢？面试官是不是要求考虑任意大的数字？如果面试官确认题目要求的是任意大的数字，那么这个题目就是一个大数问题。此时需要特殊的数据结构来表示数字，比如用字符串或者数组来表示大的数字，才能确保不会溢出。

■ 其次需要考虑各种边界值的测试用例。很多代码都包含有循环或者递归。如果代码是基于循环，那么结束循环的边界条件是否正确？基于循环的代码要特别注意开区间和闭区间的使用（也就是区分 $<$ 与 $\leq$ 、 $>$ 与 $\geq$ ）。如果代码是基于递归，递归终止的边界值是否正确？这些都是边界测试时要考虑的用例。还是以字符串转换成整数的问题为例，应聘者写出的代码应该确保能够正确转换最大的正整数和最小的负整数。

■ 再次还需要考虑各种可能的错误的输入，也就是负面测试的测试用例。应聘者写出的函数除了要顺利地完成任务之外，当输入不符合要求时，面试官还希望他能做出合理的错误处理。在设计把字符串转换成整数的函数时，应聘者就要考虑当输

入的字符串不是一个数字，比如“1a2b3c”，怎么告诉函数的调用者这个输入是非法的。

前面讨论的都是要全面考虑当前需求对应的各种可能输入。在软件开发过程中，永远不变的就是需求会一直改变。如果应聘者在面试时写出的代码能够把将来需求可能的变化都考虑进去，在需求发生变化时能够尽量减少代码改动的风险，那他就向面试官展示了自己对程序可扩展性和可维护性的理解，必定能得到面试官的青睐。如果应聘者在解答面试题“调整数组顺序使奇数位于偶数前面”时能够考虑可扩展性，他写出的代码不仅仅只是解决调整奇数和偶数的问题，还能考虑到把调整数字顺序的功能和判断一个数字是奇数还是偶数的功能解耦。这样当今后需求功能扩展要求解决类似的问题，比如调整负数和非负数的顺序、调整能被3整除的数字和不能被3整除的数字的顺序，只需要添加很少的代码都能做到，于是提高了代码的可扩展性和可维护性。

### 三种错误处理的方法

通常有三种方式把错误信息传递给函数调用者。

■ 函数用返回值来告知调用者是否出错。比如很多Windows的API就是这个类型。Windows中很多API的返回值为0表示API调用成功，而返回值不为0表示在API调用的过程中出错了。微软为不同的非零返回值定义了不同的意义，调用者可以根据这些返回值判断出错的原因。这种方式最大的问题是使用不便，因为函数不能直接把计算结果通过返回值直接赋值给其他变量，同时也不能把这个函数计算的结果直接作为参数传递给其他函数。

■ 当发生错误时设置一个全局变量。此时可以在返回值中传递计算结果了。这种方法比第一种方法使用起来更加方便，因为调用者可以直接把返回值赋值给其他变量或者作为参数传递给其他函数。Windows的很多API运行出错之后，也会设置一个全局变量。函数调用者可以通过调用函数GetLastError分析这个表示错误的全局变量从而得知出错的原因。但这个方法有个问题：调用者很容易就会忘记去检查全局变量，因此在调用出错时忘记做相应的错误处理，从而留下安全隐患。





■ **异常**。当函数运行出错时，程序就抛出一个异常。程序员可以根据不同的出错原因定义不同的异常类型。因此函数的调用者可以根据异常的类型就能知道出错的原因，从而可以做相应的处理。另外，由于显式划分了程序正常运行的代码块（try模块）和处理异常的代码块（catch模块），代码的逻辑比较清晰。异常在高级语言如C#中是强烈推荐的错误处理方式，但有些早期的语言比如C语言还不支持异常。另外，当抛出异常时，程序的执行会打乱正常的顺序，对程序的性能有很大的影响。

上述二种错误处理的方式各有优缺点。那么面试时应聘者该采用哪种方式呢？这要看面试官的需求。在听到面试官的题目之后，应聘者要尽快分析出可能存在哪些非法输入，并和面试官讨论该如何处理这些非法输入。和面试官进行这样的讨论对应聘者是有利的，因为面试官会觉得他对错误处理有着全面的了解，并且还会觉得他有很好的沟通能力。

代码的鲁棒性

鲁棒性是指程序能够判断输入是否合乎规范要求，并对不合要求的输入予以合理的处理。容错性是鲁棒性的一个重要体现。不鲁棒的软件在发生异常事件时，比如用户输入错误的用户名、试图打开的文件不存在或者网络不能连接，就会出现不可预见的诡异行为，或者干脆整个软件崩溃。这样的软件对于用户而言，不亚于一场灾难。

由于鲁棒性对软件开发非常重要，面试官在招聘时对应聘者写出的代码是否鲁棒也非常关注。提高代码的鲁棒性的有效途径是进行防御性编程。防御性编程是一种编程习惯，是指预见在什么地方可能会出现异常，并为这些可能出现的问题制定处理方式。

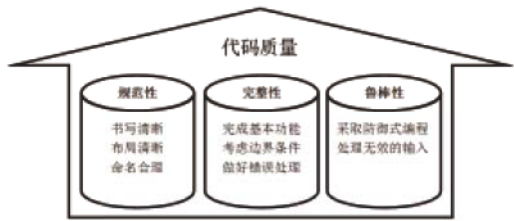
在面试时，最简单也最实用的防御性编程就是在函数入口添加代码以验证用户输入是否符合要求。通常面试要求的是写一两个函数，应聘者需要格外关注这些函数的输入参数。如果输入的是一个指针，那指针是空指针怎么办？如果输入的是一个字符串，那么字符串的内容为空怎么办？如果应聘者能把这些问题都提前考虑到，并作相应的处理，那么面试官就会觉得他有防御性编程

的习惯，能够写出鲁棒的软件。

当然并不是所有与鲁棒性相关的问题都只是检查输入的参数这么简单。应聘者看到问题时，要多问几个“如果不……那么……”这样的问题。比如面试题“链表中倒数第k个结点”，这里隐含着一个条件就是链表中结点的个数大于k。应聘者就要问自己如果链表中的结点不是大于k个，那么代码会出什么问题？这样的思考方式，能够帮助发现潜在的问题并提前解决问题。这比事后让面试官发现问题之后应聘者再去慌忙分析代码查找问题的根源要好很多。

小结

本文从规范性、完整性和鲁棒性三方面介绍了应聘者如何在面试时写出高质量代码（如下图所示）。



第一，应聘者在白纸或者白板上手写代码时要注意规范性，尽量清晰地书写每个字母，通过缩进和对齐括号让代码布局合理，同时还要合理命名代码中的变量和函数。第二，应聘者最好在编码之前全面考虑所有可能的输入，确保写出的代码在完成了基本功能之外，还考虑了边界条件，并做好了错误处理。只有全面考虑到这三方面的代码才是完整的代码。第三，应聘者要重视代码的鲁棒性，确保自己写出的程序不会轻易崩溃。平时在写代码时，应聘者最好养成防御式编程的习惯，在函数入口判断输入是否有效并对各种无效输入做好相应的处理。应聘者如果能够做到这三点，自然就能写出高质量的代码，最终通过面试拿到Offer也将是水到渠成的事情。P



**何海涛**  
思科高级软件工程师，之前先后任职于Autodesk和微软。主要关注C++/C#的开发技术，并对设计模式和项目管理也很感兴趣。

责任编辑：董世晓（dongsx@csdn.net）

# 不做只会编码的人

## 硅谷顶级产品大师Marty Cagan专访

记者 / 董世晓 陈博

Marty Cagan是享有世界声誉的产品管理专家，曾担任Netscape副总裁、eBay产品管理及设计高级副总裁。近日，记者在“PM-China首届产品经理高峰论坛”上对他做了专访，请他分享自己的产品管理历程。

### 程序员的工作

《程序员》：据我所知，您先后毕业于加州大学和斯坦福大学。这两段经历有何不同？

Marty Cagan：像这个行业的大多数人一样，我在加州大学修了计算机专业；后来我又去斯坦福大学修了一个类似于MBA的专业，那个专业的学费相当昂贵，学习的是商务方面的东西。

在加州大学，我的生活基本上都是在忙碌中度过，原因就在于我将大把的时间都花在了编程上面，没办法，这是我最大的爱好。再就是我曾经做过一段时间的网球教练，教女孩子们打网球。当教练的收入，都用来支付学费了。其他的倒也没有什么特殊的印象了，总之就是少玩多学习。

《程序员》：您的第一份工作是做程序员吗？

Marty Cagan：从加州大学毕业后，我就去了惠普实验室，做了10年程序员，主要做面向对象和基于规则的语言及工具等方面的研究。当时惠普实验室非常棒，是众多优秀程序员梦寐以求的地方。它是一个研究型的实验室，就像Google X实验室那样。再后来，我和大家一起创立了SESD（HP Software Engineering Systems Division）。

### 转行做产品经理

《程序员》：既然您在技术方面有这么好的发展，为何想到要转行做产品管理呢？

Marty Cagan：惠普实验室的产品管理做得特别好，我特别喜欢他们做事的方式，并且不想始终



Marty Cagan：打造用户喜爱的产品

是在别人的产品中做只会编码的人，我也想做产品。因此，基于这些方面的考虑，我开始在惠普实验室恶补产品管理方面的知识。此后差不多5年的时间里，我既做编程又做产品管理，这让我变得非常忙碌，但同时也乐在其中。

《程序员》：您做的第一款产品是什么？感觉如何？

Marty Cagan：我的第一款产品是在惠普实验室产生的，那是一个金融分析系统，旨在帮助金融公司分析投资损益。那时我还是一个菜鸟级程序员，只知道是按照别人告诉我的那样去做。不过，在这个过程中，虽然只能照别人说的去执行，但我确实还是从中懂得了很多事情，也可以说是作



为管理者的开门砖。

很庆幸我有一位非常好的经理，他在如何成为更好的程序员方面给了我很多帮助。要知道在美国，大学教的计算机科学与实际在公司从事计算机科学工作是大相径庭的。经理给我讲了专业编程和大学编程的诸多不同，并且让我坚信合格经理的首要责任就是帮助手下的程序员成为一名优秀的程序员，这让我立志要成为一名好经理。但实际情况是大多数经理并不关心如何帮助程序员更好地成长。不过在惠普，这被认为是所有经理的最重要的工作。我非常感激这一点，并且也致力于这样做。

可以说，在惠普的十年时间里，我学会了如何做到真正的“好”，主要是怎样成为好的程序员和好的经理两方面。

## 最好的产品

《程序员》：您认为自己做得最好的产品是什么？

Marty Cagan：这个很难回答，任何一款好的产品都不属于我自己，而是属于整个团队，是由伟大的设计者和程序员共同完成的。如果非要选的话，有两款产品让我引以为豪。

一款是在惠普时做的，名叫SoftBench。它是一个开发环境，对程序员很有用，而且我们确实是用心来做的。Softbench销售了将近20年，无疑是非常成功，也是我们当时做得最棒的一款产品。但SoftBench不是桌面软件，只是专业人员在用。

另一款是Netscape Platform。它是一款基于互联网平台的产品，很多人都能使用它。为了做好这款产品，数以百计的人为之奉献良多，我只是其中之一而已。

Netscape有很多非常聪明的人，他们都是硅谷的大牛，我从他们身上学到很多。可以这么说，在Netscape，仿佛置身于一所拥有世界上最著名教授的大学，而这些教授恰好就像是专门为你准备的一样。我每天都能感受到令人激动的现象，体验到技术的飞速进步以及客户数量和质量的惊人增长。这些感受几乎都是同时到达身体的每一个毛孔，很梦幻。一直以来，我都认定Netscape是

我工作过的最好的公司，我愿意身处于这样的团队中工作。

《程序员》：在您看来，过去十年世界上最好的产品是什么？

Marty Cagan：过去十年，好的产品不胜枚举，取得的成绩也为世人所惊叹。有些人会说Google AdWords最好，因为它为Google赚了相当多的钱。但我却想将这一票投给Facebook，因为它改变了世界上数十亿人的生活，通过它可以很方便地联系朋友和家人，这是相当了不起的成就。当然，Twitter也很出色。此外，还有很多其他很小但很好的产品，Dropbox就是其中之一，它借助云计算，提供同步本地文件的网络存储在线应用，得到了巨大的发展。总之，好的产品数以百计，但萝卜青菜，各有所爱，我只能谈自己的感受。

## 创立公司

《程序员》：再给我们讲讲您在Continuus、AOL和eBay的情况吧。都有哪些收获？

Marty Cagan：Continuus是一家创业公司，我作为联合创始人加入了这家公司。在那里，我学到了创业公司应该如何开展工作，这非常有价值，而且持续影响着我。

我在AOL虽然待的时间不长，但积累了大量互联网服务的经验。我对这个非常感兴趣，要知道AOL当时是一家数得着的大型互联网服务公司。

eBay培养了我两方面的能力，一个是如何建立新兴公司，另一个是关于产品组织，这是很特别的经验。eBay建立了让人赞叹的商业模式。在那里，我可以建立我所推崇的工作组织，这无疑是一段非常美好的经历。

但在这么多家公司工作过之后，我觉得累了：每天都是无休止的教导和谈话。

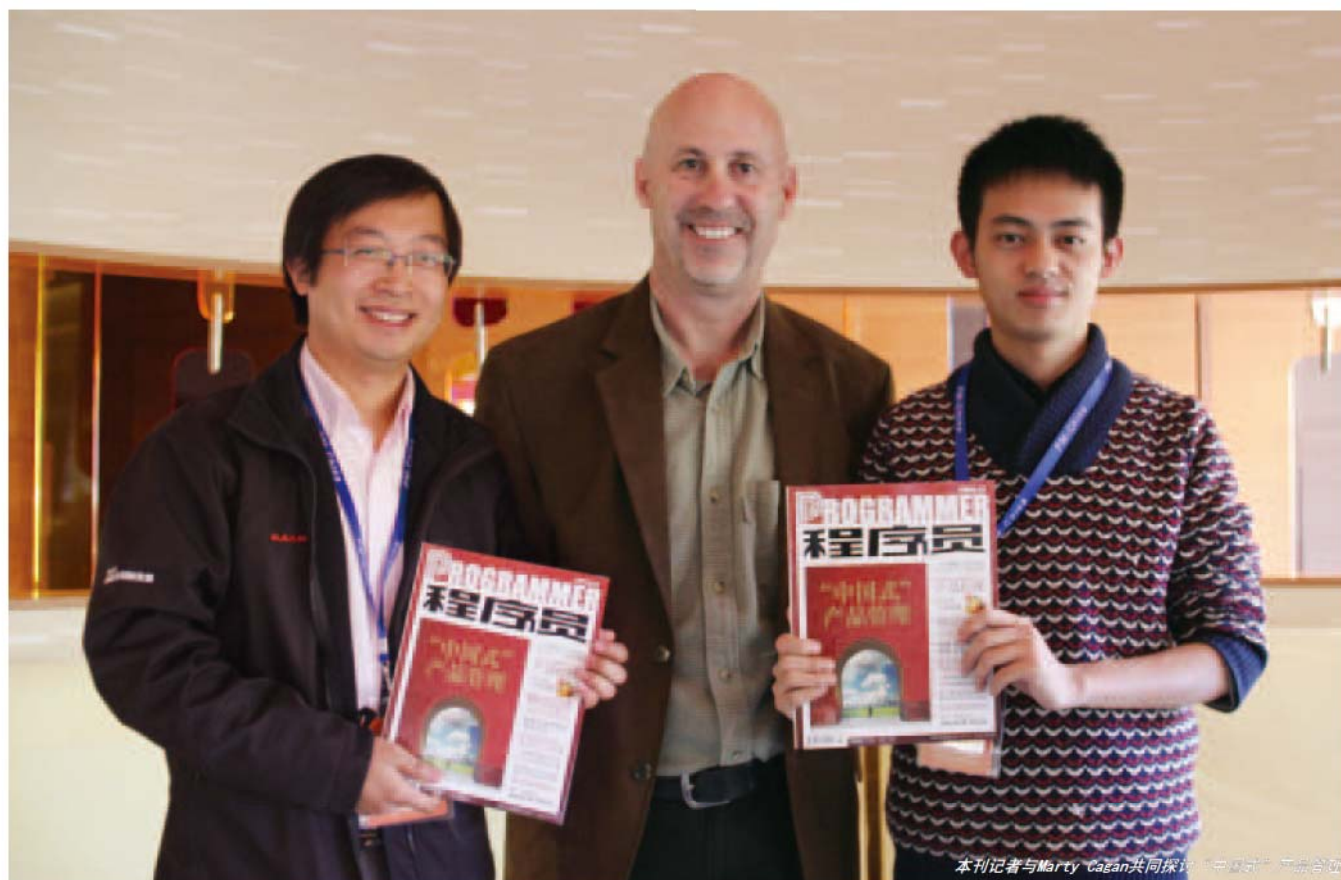
《程序员》：因此您就转而想到自己创立公司了？

Marty Cagan：是这样的，我创立了SVPG（硅谷产品集团）。

《程序员》：SVPG从事哪些方面的业务？

Marty Cagan：SVPG虽然目前只有4个人，但大家





之前曾在别的公司共事过，并且都坚信不想再为其他公司工作，而是帮助公司成长，所以我们相处得非常愉快。

我们主要是帮助新公司，以朋友的身份对其进行建议和投资，就像孵化器一样。世界上已经有很多这样的孵化器，在中国的数量也不少。硅谷有非常多的创业公司，它们急切需要得到帮助，而我们也乐于被创业公司包围，于是选址在硅谷建立办公室。我们会尽力帮助需要帮助的公司，现在同时和我们合作的公司差不多有50家。

《程序员》：那你们和Paul Graham的Y Combinator有何不同？

Marty Cagan：毫无疑问，Y Combinator是世界上最好的孵化器之一，事实上它更像是硅谷的一家公司，有大量的资金。而SVPG只是顾问，我的搭档也是在硅谷做产品管理的，因此我们没有多少钱。如果能被Y Combinator看中，是一件很幸运的事情。因为你不但能拿到投资，而且还能得到非常有价值的建议。

## 最信奉的产品管理理念

《程序员》：您最信奉的产品管理理念是什么？

Marty Cagan：这确实很难回答。我只想说两点。

一点是，伟大的产品经理知道什么是他们所不知道的。他们不认为自己知道所有的事情，不知道的事情还是占大多数。他们坚信必须聚集一起才能确定哪些是能做的、哪些是不能做的，所有这些决定都是基于坚实的证据而不是凭空的想法。

另一点是，伟大的产品经理知道他们不必是所有想法的源头，这些想法可以来自于世界上的任何地方、公司的所有员工。有时候，客户可能成为关键所在，因为他们知道你所不知道的一些事情。

《程序员》：Steve Jobs做产品时强调把自己的设计理念和想法带给用户，而不去征询他们的需求，尽管很多人不赞同，但Jobs无疑是成功的。您怎样看？

Marty Cagan：Steve Jobs以不向客户询问应该做什么而出名，这点我非常赞同。有时候，我们最大的

误会是认为客户可以告诉我们做什么，但实际上他们不能，Steve Jobs很明白这一点。但大多数人对此理解不够，认为主要任务就是向客户咨询应该做什么。Steve Jobs认为这样不可取有两大原因：第一，客户并不知道什么是可行的；第二，客户不知道什么是他们想要的，直至看到产品。因此，你并不能从客户那里获得如何做，而必须将产品展示给他们之后再收集反馈。这就是Steve Jobs做得好的原因。他不害怕与众不同是因为他知道不能事先从客户那里获得做什么，也非常清楚对手们在做什么——还在集中精力地收集用户需求信息。

## 打造用户喜爱的产品

《程序员》：您写作《启示录：打造用户喜爱的产品》这本书的初衷是什么？

Marty Cagan：我写书的最初目的就是想分享我所学到的东西。在之前工作中我学到了很多，我想找到一种方式去分享，因此离开eBay后我就想到将这些经验写下来。原本我在博客上进行了分享，但我了解到很多人更愿意通过书来了解。于是，我通过书和博客两种形式同时进行分享。我并没有想通过出书来赚钱，也没必要通过出书来赚钱，但它确实是一个很好的分享方式。

让我没有想到的是，这本书传到中国，翻译出了中国版本，让很多从未听说过我的中国人都能喜欢，确实很给力。

《程序员》：在您看来，如何打造用户喜爱的产品？

Marty Cagan：对于这个话题，我总结为十条。

- 彻底沉浸到用户中去，成为毋庸置疑的用户专家，但不要盲从于用户。
- 区分愿景和假想的不同，通过与用户面对面交流来验证愿景，虚怀若谷。
- 成功的前提是你必须知道：要解决什么问题？为谁而解决？如何解决？
- 拥抱用户体验，因为用户体验通常比开发更重要、更困难。
- 快速过滤糟粕，要铭记：产品部门的核心职责是进行产品探索。
- 确保坦诚合作。

- 快速改进，向着卓越的方向不断优化产品。
- 敏捷化，通过增量和迭代的方式构建和发布，务必确保产品是可信赖的版本。
- 追求卓越，保持高标准的同时注重每个细节。
- 打造、培育和鼓励创新文化，为其优化组织和流程。

## “中国式”产品管理

《程序员》：在您看来，中国产品管理是个什么情况？请给中国的产品经理们一些建议。

Marty Cagan：中国实在太大了，对产品经理也很难概括出通用的说法。但基本会分为两类。一类已经很像硅谷的产品经理，他们有非常成熟的经验。另一类像是在巴西工作时遇到的一样，还处于“学习产品是什么”的阶段。同样作为新兴市场，中国和巴西的产品管理起步时间都不长，所以我建议更多的是要关注产品管理本身，不应该把产品管理者定义为项目经理。中国拥有数量庞大的优秀程序员，这是事实，但在产品管理方面薄弱也是不争的事实。

很期待中国的产品人员能够为中国创造出更多伟大的产品，这样不仅有利于中国，也会造福世界。

《程序员》：在您看来，产品管理有何趋势？

Marty Cagan：在我看来，有三种趋势将在世界上流行很长时间。

最大的趋势应该是Lean Startup（精益创业）。Lean Startup是一种非常好的思想，各方面的数据也证明了它确实非常快。

其次是敏捷方法。很奇怪的是，中国的很多公司竟然没有听说过敏捷，听说过的只是单纯地在学习，切实使用敏捷的是少之又少。这不是一个好现象。在我看来，未来三年，敏捷将对世界产生巨大影响，绝大多数团队或者程序员将会使用它，中国在这方面不应该被落下。

最后一个趋势是设计，中国技术人员是时候认真对待设计了。📌



# 产品经理的时间管理

文 / 汤圆 老马

产品经理总是为各种会议而分心，以至于自己的工作都没有太多时间处理。产品经理如何做好时间管理？且看本情景小品。

## 征用归来

“小谢和亮仔呢？又去开会了？”周扬刚走进办公室，就看到他们的座位空空如也，于是问小娟。

小娟早已习惯这种情况，不在意地说道：“又被市场部叫去了，说有件事情需要确定一下。”

周扬没说什么，就坐下忙自己的事了。自从进入执行阶段后，他俩要参加的会就多了起来，什么类型的都有。简单来说，只要和产品沾点边的会，不管这个会到底和他们有多大关系，都得参加。

没一会儿，他俩就回来了，一脸的无奈，刚进门，亮仔就抱怨起来：“这会开的，推广部门研究怎么写产品软文也让我们去，一群人扯了半天也没个结果，虽说我有一定的文学造诣，但我擅长的是诗词歌赋，这类新闻稿式的说明文确实无甚研究啊。”

“拉倒吧，这你今天的第几个会了？”周扬问道。

“第三个了吧？”亮仔想了想，然后说道。

“不对，亮仔，应该是第四个了，你忘了早上10点多的时候，研发部的还来这里和你开了个10分钟的小会呢。”谢韵琳提醒亮仔他的计算标准有误，不但要计算到别的部门开了几个会，还得计算别的部门来产品部开了几个会。

“嗯，那就是4个了，一天到晚尽开会，手头的事一点都没做。”亮仔叹道。

“在这个阶段，产品经理开会多在所难免，很正常，习惯就好了，呵呵！”我宽慰亮仔道。

“我看我快熬成浆糊了，正准备做点事，通知开会，思路一下子没了，一会儿一个会，脑袋里现在还乱七八糟的呢，不去开还不行，人家说了，你是

产品经理，不参加怎么行呢。去参加吧，自己的事没时间做，不去参加吧，得罪人。”看来我的话不但没让亮仔感到宽慰，反而更让他牢骚满腹。

“是不是所有公司的产品经理都是这样啊？”谢韵琳了解一下有没有同命相连的战友。

我想了想，说道：“规模小点的公司可能还好些，一方面是因为只有为数不多的人在维持着公司的运作；另一方面，也和公司的发展有关，当公司规模还比较小的时候，每个人都可以参与到公司运转中，包括每次的会议。”

“那你意思是说咱会议多，说明咱是大公司，是正常现象了？呵呵！”亮仔有些自嘲地说道。

“必要的会议是必须参加的，但现在的情况是咱参加了很多对产品经理来说无足轻重的会议，这就有点不正常了。”周扬边掏烟边慢条斯理地说。

## 会议分两种

大家立刻明白，要内部会议了。

“扬哥，那你说应该怎么办？”亮仔竟然没去拿烟，看来今天的会把他折腾得够呛。

“我只说说如何避免那些本来不需要开的会。分两种情况。第一种情况：产品经理自己开了很多无足轻重的会。之所以出现这样的会议，是因为产品经理太过于敏感了，或者就是产品经理对于自己的团队能做好某件事情并没有足够的信心。比方说，研发部正在按照你的PRD在开发产品，但你总是琢磨他有没有完全理解了某个功能啊，他们能不能真的按照我的想法去做呢，他们在做的过程中到底认不认真啊，甚至有时候你都在想他们现在是不是



在做啊。因此，你总是处于一种焦虑之中，对研发团队能否做好这个产品心存忧虑。因此，要么你时不时地去看看他们做什么，要么就是隔三岔五地开个临时会议。其实在大部分情况下，咱真没必要去开这些会，公司有规范，有制度，有流程，你完全可以通过必要的制度掌控产品的进度。”

亮仔笑了笑，说道：“扬哥，你就放心吧，我绝对不会出现这样的问题，我对咱的研发部那是一百个放心，不过就是小谢……”说着，他就转向谢韵琳，装着批评的语气说道：“小谢啊，既然扬哥说到这儿了，我就得批评你几句了，你看看这段时间，你总是问我研发部做得怎么样了，今天知道问题了吧，这是对研发部的伙计们不信任啊。”

谢韵琳也不好意思地笑道：“那怎么来解决呢？”

“其实就是心理作用，调整好自己的心态就可以了，要告诉自己，我已经把这个产品跟我的团队说得很清楚了，他们知道我想要什么，除非有什么大问题发生。只有这样，你才会相信一切尽在你的掌握之中。小谢有这样的心态其实也正常，因为是刚刚做产品经理，第一次负责产品的全部过程，许多事情并不熟悉，时间长了，接触得多了就好了。”

周扬继续说道：“第二种情况就比较普遍了，就是我们现在遇到的，业务部门要举行一个会议，非得让你去参加，咱们先分析一下，为什么会出现这样的情况。”

“我是这么认为的，一方面是因为咱公司刚实行产品管理，许多部门并不熟悉流程，现在都处于一个磨合的过程中，可能在他们的想法中，认为只要是和产品相关的事务，不分大小，都应该知会产品经理。另一方面，咱自己可能也缺乏对某个会议重要性判断的经验，直到会开始了，才知道这个会是可来可不来的那种。”刘宇非常认真地分析道。

周扬点点头，说道：“刘宇的分析很有道理。对第一种情况，没有什么好的改善办法，只能是不不断加强他们对产品管理的认识，让他们知道产品经理在产品上并不是事事都要操心的，一些事情完全是由他们来决定的，尤其是在战术执行阶段。这需要时间，咱们一点一点来。主要看看第二点，这是咱应该做到的：如何判断这个会议是否需要参加？”

还真有些难，大家想，但凡要开会，举办方肯定



认为这件事情重要，否则谁没事老开会玩啊。但问题就出现在这里，举办方认为重要的、必须要开的会议，但参会方不一定也这么认为。这就容易出现矛盾了，对于产品经理来说更是如此。

## 两个窍门

“怎么判断？”我问周扬。

周扬笑了笑，说道：“我有两个小窍门，当然，这只是我多年在文山会海中体会出来的，不敢说百分之百吧，还是有一些借鉴意义的。

第一个小窍门，就是在接到会议通知时，问清楚都有哪些人参加，如果参加的人大部分都是一般的员工，那么这个会基本不用去参加。原因很简单，一般员工肯定没有权力就某个影响产品大局的事情开会，而产品经理要参加的会，肯定是和关键产品事务有关的会议，不是和咱们平级的部门经理，就是更高一层的各位老大们。大家想想，前期咱们参加的会议是不是基本是这个特征？”

细细一想，果然如此，我参加了几个会议，似乎每次都有韩老大。

“如果是这种会议，那咱们就不用去参加了，因为也不可能讨论什么关键性的问题。”周扬又点上一支烟，然后继续说道：“第二个小窍门，如果某个会议是临时通知你立刻参加的，那么这个会议基本上可以推辞掉，因为如果是非常重要的，事关整体或者关键问题的会议，肯定会提前通知，最起

码也得提前一天，有的可能会提前好几天。比如咱的评审会，就提前三天通知。这是因为参会的人都很关键，必须要提前通知，做好必要的准备，而这种临时通知的会议，肯定是偶尔想起有什么事情需要和咱们沟通，于是决定开个会。放心，这种会不会讨论什么大的问题，因为只要前期的工作到位了，临时出现关键性问题的可能是非常小的。当然，也不能说完全是这样，还得看通知的人是谁，如果就是一个普通员工来通知的，那么肯定不会很重要，如果是公司有一定级别的人物临时通知的，那么这个临时的会议肯定也非常重要。不过这种情况不多，重要的会议肯定会提前走流程的。”

周扬说完，我不由想起了郭姐姐，好几次她临时通知周扬或者我们开会，果然都是比较关键的会议。

“真是长知识了，就连咱们如何参加会议都有这么多门道！”亮仔感触颇深地说道。

“看起来是如何应对各类会议，其实体现出来的就是时间管理的思想。”刘宇在一旁又说道。

周扬点点头，说道：“就是这个意思，咱们一天的时间是有限的，但是要做的事情却总感觉是无限多的，一天下来累巴巴的，却感觉没做什么真正有用的事情，究其原因，还是因为我们没有区分清楚事情的重要紧急程度。”

“啊，周经理说这个我就知道了。”苏勇一直在默默地听，但一听到周扬这么说，马上说道：“就是按照事情的重要、紧急程度划分级别的矩阵吧？”

我心想，时间管理的观念现在看来也算是众人皆知了，那个按照事情重要、紧急程度划分事情级别的标准也可谓深入人心，满大街的人都知道。

		急迫	不急迫
重要	A	立刻处理	B 根据你的愿景和价值来组织这些活动
	C	不要把所有急迫的事都视为重要的事——好好选择并加以评估	D 尽量减少或予以剔除
不重要			

时间管理矩阵

时间管理矩阵

虽然如此，阿泡还是决定把这个时间管理中最经典的矩阵再画一遍，大家再回顾一下吧。

“其实说到底，时间管理就是在强调两个概念：效率、效果。

什么是效率？效率就是指能够产出、不会浪费时间的能力；效果则是指能够针对正确的事情排序并进行工作的品质。

也可以这么说，时间管理的目标就是在‘效率’和‘效果’上达到最适合的平衡。”刘宇又要准备非常有理论、非常有逻辑的开始由浅入深地讲时间管理的東西。

亮仔赶忙制止，说道：“刘爷，理论的东西咱就别说了，说点可操作性比较强的吧，就和刚才咱们聊开会那样的，有什么方法或者窍门什么的。”

刘宇被他这么一搅和，自然没有什么兴趣继续往下说了，坐在那里一言不发了。

“就你事多，你要是知道时间管理怎么做，还会被麻烦成这样？理论的才是能够引起思考的，什么事都想找现成的方法，那咱们这产品经理未免也太好做了吧！”周扬虽然比较照顾亮仔，但在原则问题上，绝不会出现偏袒的情况。

一是一，二是二，公是公，私是私，这也是我最佩服周扬的一点，能做到这一点，在中国的公司里，不容易啊。

亮仔不好意思地笑了笑，然后对刘宇说道：“刘爷，我就是开个玩笑，你看你还当真了，你不是经常说，产品经理要大度，你现在就大度一些呗，您说着，我听着。”

说完，又是憨憨一笑，我知道，刘宇其实也没有生气，这就是他的性格，太认真，心里有想法一定要说出来。

刘宇也冲他笑了笑，说道：“那我就按照亮仔的建议，直接说说我的经验吧。”

其实想想，很多事情都是这样，各退一步，万事无忧，产品经理要面对的各类角色太多，很多时候都会形成一种对立面，在不涉及原则的情况，



产品经理要学会先往后退。

“我认为，时间管理对于产品经理最大的启发就在于，如何变‘救火’为‘预防’，我敢说，越往后走，救火的事情会越来越多，到那个时候，如果咱们没点方法来处理这些事情，估计要比现在累10倍。”

## 产品经理要会说“不”

我不知道其他人有没有这样的感受，我深有同感，现在几乎所有的产品经理都在扮演“消防员”的角色，到处救“产品的火”，一天下来，累得要死，但是效果却不尽如人意，还往往把重要的事放在了一边，被高层认为“既没有效率，又没有效果”。因此，我不住点头。

“要改变这种情况，唯一的方法就是把‘救火’变成‘预防’。”刘宇继续说道：“其实，在日常的救火工作中，你会发现，有许多火是同一原因引起的，这个时候，你就要想一些办法了。最常见的方法就是‘压缩法’，把所有相同原因引起的问题整理出来，压缩成一个问题，然后提交给相关部门，这样，一旦再遇到这些问题，他们会通过‘解压缩’的方法来自我救助，而不是首先想到去找产品经理。”

“这有效吗？”亮仔很关心这个方法的效果。

“嗯，这么说吧，据我个人的经验，压缩法可以为产品经理节省出日常工作时间的40%。”

“那其他不能压缩的事情呢？”苏勇也问道。

“至于其他无法压缩的事情，还得遵循刚才你说的那个事务划分级别的标准，先确定级别，然后考虑怎么去做。”

“那有没有一个可行的方法呢？”显然苏勇也知道那只是一种思想，如果有一些方法能够支撑或许更好。

刘宇想了想，说道：“退一万步讲，如果咱们不能确定这些级别，或许有一个笨办法可以试试。”

“方法不怕笨，只要能解决问题就行。”周扬在旁边说道。

“还是我个人的经验啊。我喜欢的一种方法是每天为这些事情留出2~4个小时。在这个时间段

内，我只把精力放在我认为所有和产品有重大关系的事情上，比方说专门处理各类业务邮件，或者完成自己应该做的事情，绝不去参与一些临时性的事务或者被这些事务所干扰。当然，可能这是一种强迫性的工作，但是时间长了，你就会逐渐发现哪些事情是真正重要的，然后就会有意识地知道如何来排级了。”

“我明白了，这个方法就等于强迫自己留出一些时间来专门处理一些事情，但很有可能在一开始的时候，许多事情并不是自己想得那么重要，但是时间长了，就会知道了，对吧？”亮仔说道。

“是的，要不我说这是一个笨办法呢。”

“不笨，正适合我现在的情况，我现在就是太容易被其他部门的事务所干扰了，从现在开始，我就得用这个方法了，这叫什么，对了，就叫‘焦点时间法’吧！”亮仔说完，自己都乐了。在一定的时间内专门处理焦点事务，这个名字起得很贴切，很形象嘛。

我们正说着呢，亮仔的座机响了，亮仔赶忙接起来，具体内容不知道，但是听亮仔说似乎又是要他去开一个会。

最后，只听亮仔和对方说道：“兄弟，现在是我的焦点时间，不好意思啊！”

说完，就把电话挂了，转头对我们说道：“销售部的电话，让我和小谢过去和他们销售部交流一下产品，不过我现学现卖，这种会就没什么必要了，作为产品经理，咱也得学会说不，是吧！”

其实仔细想想，亮仔说得其实很有道理，产品经理为什么总感觉累？在很大程度上就是因为不会说“不”，或者不好意思说“不”。就从很小的一个开会的事情上就能看出来，要想工作有效率，有效果，作为一个产品经理，“不”，就要大胆地说出来。P



本文节选自《YES! 产品经理》一书第93问，该书已由电子工业出版社博文视点公司出版。特此感谢博文视点公司授权。

# 敏捷团队中 可视化的任务分配管理实践

文 / 许正华

本文分析了任务分配中的若干挑战与根本目标，并基于人力发展的一般模型，提出了一种可视化的任务分配深度管理之道。

任务分配是软件项目管理的重要组成部分，其合理性很大程度上影响产品的最终成败，因为软件生产的根本问题是人的问题。传统意义上，管理者希望为任务找到合适的主人——人是被当作工具来使用的。敏捷告诉我们，“人”才是最重要的。

## 软件任务分配的传统挑战

- 挑战一：研究性工作由少数骨干员工所“垄断”。这些研究性工作包括对关键技术的学习、原型开发、架构设计等。垄断乃玩笑话，实则是由员工的个人素质与客观环境（任务重/时间紧）共同作用的结果。
- 挑战二：辅助性工作“无人问津”。辅助性工作是指项目中的脏活，例如性能测试，它们往往费力不讨好，容易使人产生负面情绪。这些工作经常是由团队中的新人来承担。
- 挑战三：开发任务量分配不均，即“巨人症”。团队中能力出众、责任心强的员工往往承担了大量的复杂性开发任务。这种分配只能造成强者愈强、弱者愈弱的结局，甚至危及团队的稳定。
- 挑战四：开发内容分布不均，即“偏食症”。随着项目的深入，每个人所承担的任务过于单一，一个人所接受的任务往往是自己所相对熟悉的。这样做的危害是团队内很容易出现知识壁垒，进而使成员间无法开展积极的合作，产品的最终质量难以保证，增加了项目实施风险。

## 敏捷开发中的任务分配

### 计划经济与市场经济

任务的自主管理是敏捷团队提倡的以人为本精神的重要体现。有人认为自主管理就是简单地根据自己的兴趣自由选择，是敏捷开发所特有的，我认为这些观点都是不全面的。我们可以从当年对“计划经济”与“市场经济”的探讨中吸取一些经验教训。资本主义与社会主义的区别并不在于计划经济与市场经济；传统的软件开发与敏捷的软件开发的区别也不在于团队成员能否自由地选择任务。任务的自主管理对任何软件过程都是有益且可实施的。

任务的自由选择并不等于绝对的自由主义，而应建立在团队共同目标基础之上。自由选择只是一种手段，而不是目标。隐藏在它背后的真正目标应该是让每一个团队成员有机会得到均衡的发展，因为均衡发展最符合敏捷开发的需要，也符合每一个团队成员的长远利益。

### 深度的任务分配管理

施耐德（Schneider）将公司文化分为四种类型：协作型、控制型、成长型和竞争型。敏捷适合于协作性与成长型的文化。

以时间为度量对每位团队成员所承担任务及其所处状态进行监控可以有效控制项目进度。这种管理对构建一个敏捷的团队是无效的，无助于解决软件任务分配中的传统挑战，因为它的核心目标是过程。



施耐德文化模型强调敏捷是以人为导向的，而不是为了简单地对过程进行控制。现实需要一种以人员协作与人员发展为目的的深度任务分配管理。

## 人力发展二维模型

我们可以将高技术人力的发展简单地归纳为知识与技能两个方面。知识强调理论，技能强调实践。知识与技能的关系可以用“微妙”来形容，因为随着讨论主体的不同，两者之间的关系也不尽相同。对于低技术含量的工作，两者的关系并不是十分紧密：虽然学习知识对于掌握技能有所帮助，但技能可以脱离知识而存在。例如，一个拥有某项高超技能的工匠，可能对其背后的原理知之甚少。对于高技术含量的工作，两者却紧密绑定在一起：你甚至很难分清究竟谁是因、谁是果，因为它们早已发展为互为因果的共同体。这个时候，再简单地讨论“谁是最初的因、最初的果”已经没有什么意义。

基于该模型，以技能为X轴，知识为Y轴，可以将某人在项目组内承担的所有任务投射到该坐标系。理想情况下，任务应该是均匀分布的（如图1所示）。

基于软件开发的特点，可以将人力发展二维模型进行进一步演化（如图2所示）：在技能方面，细分为研究、开发与其他。在知识方面，可以按功能进行划分，也可以根据项目特点采取其他标准。

## 可视化任务统计与案例分析

“可视化”与“持续化”是敏捷项目管理中的两个重要理念与方法。我们将其应用到任务的深度管理实践中。

基于人力发展的二维模型，我们设计了一组（共四个）数据报表来展现并追踪每一位队员的任务分配与完成情况。

■ 研究任务表（技能）追踪队员所完成的研究性任务点数，并计算均值。准确地识别与定义研究性任务是完成该表的关键。

■ 开发任务表（技能）追踪队员所完成的代码行数。代码规模是衡量代码复杂度的一种最简单也是最直观的方法。

使用代码行数这个度量需要注意两个问题。

（1）区别功能代码与非功能代码。非功能代码包含了代码框架自动产生的代码与测试代码。

（2）代码行数并不能准确代表程序的逻辑复杂度。“圈复杂度”（Cyclomatic Complexity）是目前被业界普遍用来衡量程序逻辑复杂度的度量。

现代的集成开发环境都可以对这些基本的代码复杂度度量进行计算，例如Visual Studio 2010。为了简便，我们目前只追踪代码行数。

■ 辅助任务表（技能）追踪除了研究性任务与开发性任务以外的所有任务。辅助性任务并非无足轻重，也并非轻而易举。完成它们往往需要更加专业的技能。例如，各种各样的性能测试就对测试人员的技术、经验提出了很高的要求。因此，所谓“辅助”任务其实是“名不副实”的。

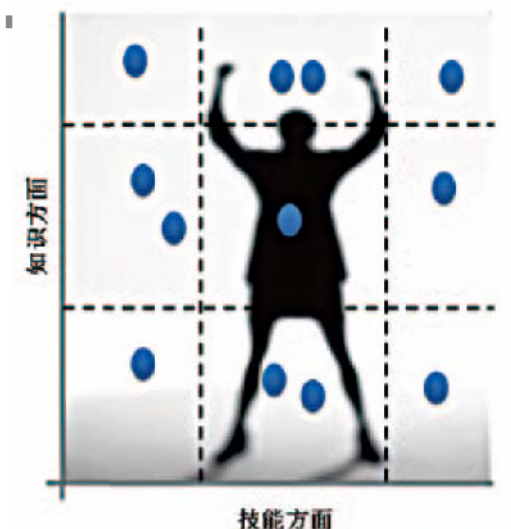


图1 任务均匀分布

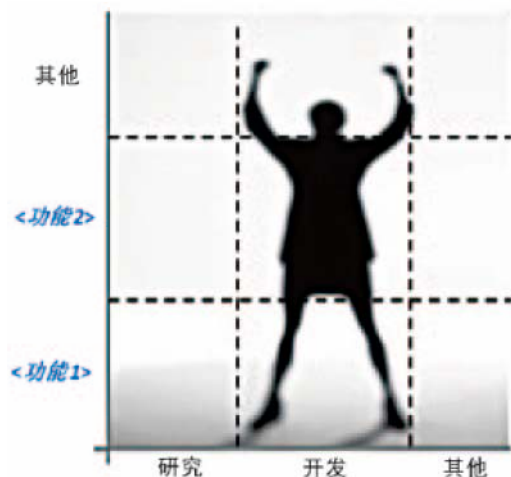


图2 进一步演化的人力发展二维模型

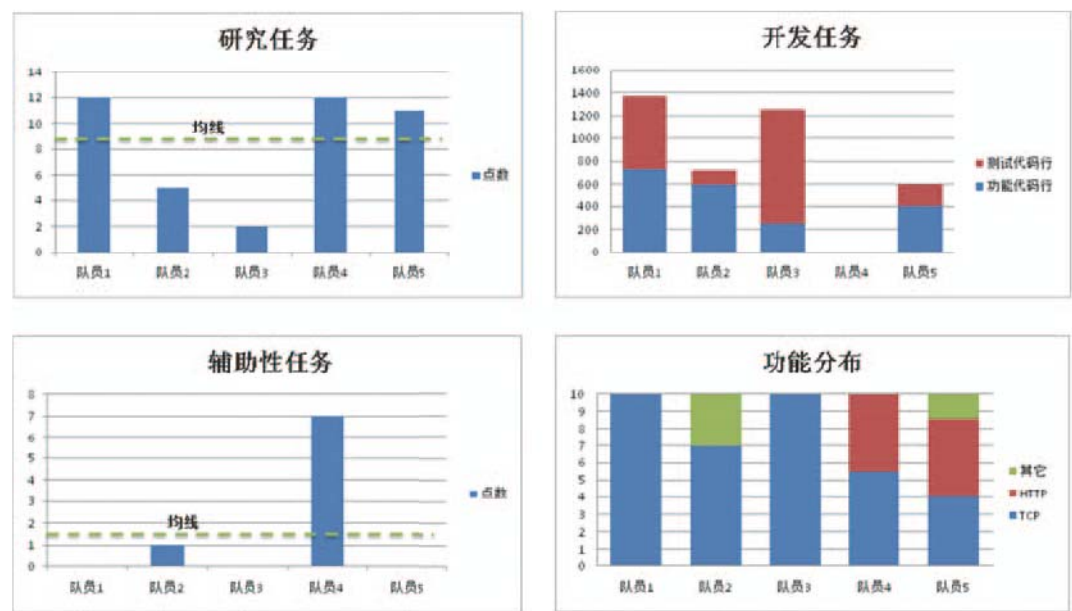


图3 数据采集示例图

功能分布表（知识）反映了每个队员所完成的各个功能方面占总完成功能的比重。每个团队需要根据自己项目的特点对功能方面进行划分。建议功能方面的数量维持在3~4之间。

功能分布表与上面三个表有以下两点不同。

- （1）维度不同。前三个表都与人的技能有关，功能分布表与人的知识有关。
- （2）数值类型不同。前三个表中的数据都是绝对值，功能分布表中的数据是相对值。

数据采集

本案例数据采自一个真实的敏捷开发团队。团队包含五位开发人员。数据采集时，项目总共执行了两个月。如图3所示。

数据分析

我们可以根据问题的数量与严重程度将队员的健康状况分为若干等级。问题严重程度分为“一般”（无标记）与“严重”（标记：【+】）两个等级。本案例中健康状况共分为以下三个等级。

- 健康：各种任务分配均匀，没有明显的问题。
- 亚健康：一个或两个指标有问题，且严重问题不超过一个。
- 不健康：两个以上指标有问题或严重问题超一个。

根据上述原则，该团队有一名健康队员，三名亚健康队员和一名不健康队员，如下表所示。

队员标识	健康等级	详细描述
队员 1	亚健康	【+】增加 HTTP 相关的任务所占的比重
队员 2	亚健康	适当增加研究性工作所占的比重 【+】增加 HTTP 相关的任务所占的比重
队员 3	不健康	【+】增加研究性工作所占的比重 增加核心开发任务所占的比重 【+】增加 HTTP 相关的任务所占的比重
队员 4	亚健康	【+】增加开发性任务所占的比重
队员 5	健康	目前为止，任务比较均衡

总结

我们以商业化的敏捷项目管理软件所提供的数据为基础，利用基于人力发展模型所设计的工具对任务分配进行深度的分析，并将结果与团队成员进行及时的分享，收到了良好的效果。需要特别说明的是，该方法的设计与应用不是为了考核员工的绩效与能力。

目前，市场上的敏捷项目管理软件只能完成比较初级的任务管理功能。为了让任务管理真正服务于建立高效的敏捷开发团队，需要得到商业化工具更加有力的支持。P



许正华  
目前就职于CA Technologies中国技术中心，从事数据备份和高可用企业软件方面的开发和研究，并专注于软件工程、知识工程、团队文化等领域的研究与实践。

责任编辑：董世晓 (dongsx@csdn.net)



# 我眼中的2011年



钟文昌  
知名Android培训专家

## 智能手机的起飞之年

### 生活

2011对我来说是特别的一年，一个人来到陌生的北京，要适应气候、人文环境及风俗民情的差异；不过我也深刻体验到浓厚的人情味，北京人的直爽、干脆、大方深深吸引了我。

11月底的第一次降雪，让我兴奋地大叫了起来，毕竟雪从天而降这一幕对于向来在南方地区生活的人而言，是多么新鲜的事。

北京干燥的天气、温暖的人情、拥挤的交通，加上听不太懂的普通话，还是让我爱上了北京。我喜欢北京的生活，毕竟大家都是中国人，没有民族的隔阂，同文同种的血缘关系让我在寒冷的冬天感受到北京的温暖。但我还是好想念台湾的珍珠奶茶跟铜锣烧啊……

### 产业

2011是智能手机的起飞年，这一年，智能手机如烈火燎原般侵入你我的生活，爆发式的成长曲线超乎众人的想象。智能手机俨然成为我们日常生活中不可或缺的必需品，在全球手机市场的份额不断扩大，甚至是倍数成长。

面对着智能时代的来临，所兴起的微型创业模式给我们指引了一个明确的方向。App时代早已来临，IT产业已由硬件走向软件，而中国的市场亦由山寨转向品牌，数家芯片设计公司在中国市场这波智能换机潮中经历由盛而衰，深圳山寨机业者也经历着阵痛的转型期，甚至倒闭。智能时代的洗牌，瞬息万变的市场，让我认为下一个爆发式成长的市场会在医疗、汽车及家庭数码三大市场，医疗市场的兴起可能会

落后于欧美，而中国扮演全球最大电视生产及销售国，有可能引领家庭数码风潮。智能时代的重点在于应用结合，就是结合多种既有的应用及服务来创造多元的商机。

智能时代需要软硬件的高度整合、零组件及操作系统的协调搭配，大陆+台湾=智能时代下的双赢模式。以Android为例，它俨然是软件工程下的产物，结合软件工程、面向对象的编程方法以及通信及互联网相关的背景知识而成的一大产物。21世纪是中国人的世纪，中国庞大的市场优势，节节上升的竞争力及研发能力，加上IT产业的经验、创新及管理能力，绝对有机会主导未来智能时代的发展。

### 教育

IT产业在中国快速崛起，但中国当下却面临人才缺口，这个问题的症结在于中国填鸭式的教育方式。欧美开放式的思考模式造就了其较具创新力及想象力丰富的人才。App时代所需要的就是天马行空、白日做梦及爱幻想的人，因为App需要的是创意，胡思乱想的点子都可能成为下一只“奋斗的小鸟”，这也是微型创意时代对中国新生代程序员所带来的冲击。

开源的Android是我们进入智能开发及微型创业时代的垫脚石，而IT产业日新月异，在智能移动开发时代，唯有掌握过程及方法才是关键，因为结果仅是表象，而方法却是大同小异。专业能力并非一天养成，长城亦非一天建起，唯有积累扎实的研发能量，才能够在微型创业时代发挥中国的微型创意。P



马超  
微软最有价值专家 (Windows Phone方向)

## 感恩2011，一个典型程序员的成长轨迹

在我看来，我的人生将分为儿童时期、学生时期、技术人时期以及社会人时期，2011年对我来说就是这么一个转型的时期。

在2011年之前我一直作为一个程序员，一个非常纯粹的程序员，在我的世界里我总觉得“万般皆下品，唯有技术高”。


记得在我大学毕业时，作为一个刚入行的程序员和所有其他刚入行的朋友一样，满怀激情，有冲劲更有闯劲，但缺乏对于问题的认识能力。其实这是程序员成长的第一步，即：提出问题。当时一个很偶然的机会，让我从百度中搜索到了CSDN并在论坛上发言且最终获得了解答，从此我基本上把所有的业余时间都放在了CSDN的论坛及博客中。那段时间也是我提高比较快的一个阶段。

在CSDN论坛中遇到和看到的都是实际项目中所遇到的问题，所以接触得多了，有些问题也就可以独立回答了，这也是成长中的一步，即：理解问题与解决问题的能力。这个时候我成为一名CSDN社区技术专家、移动大区版主以及微软最有价值专家。

但2011年亲人的突然离世让我感到了肩头的责任，让我明白了生活中除了有工作要做，还有家人要照顾，爱人的怀孕、新生命的到来也让我真切感受到了“不养儿不知父母恩”的含义。

同时，我认识世界的角度也有所变化，我不再认为只会分配任务而不做实际工作的leader是一个蛀虫，不再认为业绩惊人的销售只是靠着喝酒去谈生意。我开始思考一个程序员的转型，并在博客上分享自己的感想。我再也不认为管理只是些简单的理论，而是对于人生阅历的升华。

2011年的我再次翻开了《Java编程思想》、《设计模式》，我开始明白代码不能只看逻辑，更要领会大师的思想，如果有人说IT业变化太快我能理解，各种编程框架语言真的是你方唱罢我登场；如果有人说IT业其实没有变，我也能理解其实变的只是表面，编程的思想其实没有改变。

2011年是我，一个典型的程序员到了一个转型年龄所经历的典型一年，在这里愿各位也愿我自己能一路走好，越走越好！



吴梦华  
互联网创业者

## 云时代快步向前，社交持续升温

对于互联网创业者来说，2011年国内最让人兴奋的当属腾讯平台开放及其引发的大型网站集体开放的连锁反应。虽然平台大佬们更为关注开放的整体成效，但第三方登录才是其中最具有价值的干货。几乎是一夜之间，中小网站们就迎来了这种OpenID技术的春天。2011年前还饱受新用户注册问题苦恼的小网站们，至此一举拥有了等同于QQ、新浪等巨型网站的用户基数。剩下的问题，只在于网站内容本身。

一方面，在给中小网站带来实惠的同时，OpenID的广泛应用将实质性地促进互联网数据和服务互通，从而为云时代扫除一个重要障碍。另一方面，Google App Engine如约去掉了Beta字样，进入正式运营。这不仅是Google对自家产品的信心，更是云时代的又一个里程碑事件。PaaS云计算平台的成熟将进一步把互联网程序员分裂为操作系统和应用两个阵营。前一阵营深入底层钻研维护Matrix般深邃的系



统，后一阵营则专司创造运营用户可见的产品。以往争论不休的服务器架构、数据库底层乃至LAMP还是.NET，或终将成为历史。

2011年国内互联网的另件大事是微博的空前成功。老牌网站新浪通过微博一举翻身，正式杀人互联网Web 2.0战场。虽然有人批评微博缺乏盈利模式，但这并不影响新浪睥睨追赶者的好心情。被新浪微博郁闷的不仅有腾讯和搜狐，还有前两年大红大紫的开心网。这匹2009年忽然杀出的黑马，曾一度以垄断国内社交服务的气势直逼腾讯，却没想到最终被新浪抢了风头。这中间有国内互联网的生态因素，但不够开放和创新不足才是真正内伤。

反观新浪，不仅在微博开放性上做足工夫，更模仿Google引入了SAE云计算平台为第三方开发者提供便利。这种用户、开发者两手都硬的做法让2011成为了不折不扣的“新浪年”，连大西洋彼岸的老师Twitter都自愧不如。Twitter仍在快速发展，但似乎已经没有了赶超Facebook的那股气势。后者在2011年轻松收获

8亿用户，并预计会以超过千亿的市值于2012年上市。

社交媒体和社交平台之争在国内外截然不同的结果表明：以领域优势判断优劣并无意义，优秀的公司自然会将触角延伸，问题只在于是否秉持足够灵活开放的态度。从这个角度上讲，互联网另一重要领域——搜索引擎的巨头们正在遭遇前所未有的挑战。无论是国外的Google还是国内的百度，都觉察到了广告收入模式之于虚拟货币发行权的小巫见大巫。向社交转型，成了搜索引擎巨头们2011年的年度大考。Google终于推出了Google+，并已收获数千万用户，但前途依然未卜。而百度似乎仍然沉醉于独霸搜索市场的好处迟迟不能下定决心。然而，腾讯和新浪已经开始瓜分明天，框计算是否还能从容应对？

2012年的钟声即将敲响。当我们回顾这个充满电商和团购喧嚣的2011时，请不要忘记，互联网可能正在孕育下一阶段的实质性爆发。P

## 寻找机遇 创造未来 CSDN 热门职位全新推荐

### 用友软件

诚聘以下职位：

**UFIDA 用友**

#### 一、Java 开发工程师

岗位要求：

1. 两年以上 Java 开发经验；
2. 熟悉 SQL Server 或 Oracle 数据库；
3. 熟悉 Eclipse 开发工具；
4. 熟练掌握 SQL 语句，能够熟练进行数据库设计。

#### 二、.NET 开发工程师

岗位要求：

1. 两年以上 .NET 开发经验，具备一定分析设计能力；
2. 良好的 C# 代码实现能力，熟悉 ASP.NET 开发技术；
3. 掌握面向对象的设计方法和设计原则。

简历投递：rdzhaopin1@ufida.com.cn

### 上海纵游网络技术有限公司

诚聘以下职位：

**DeNA China**

我们正在全力招聘中国人才，打造一个全新的中国团队，致力于中国智能手机游戏平台的构筑和运营。愿意加入我们，一起构筑世界著名的智能手机游戏平台吗？

Android/iPhone/Web application，更多职位招聘信息请登录公司主页进行查询

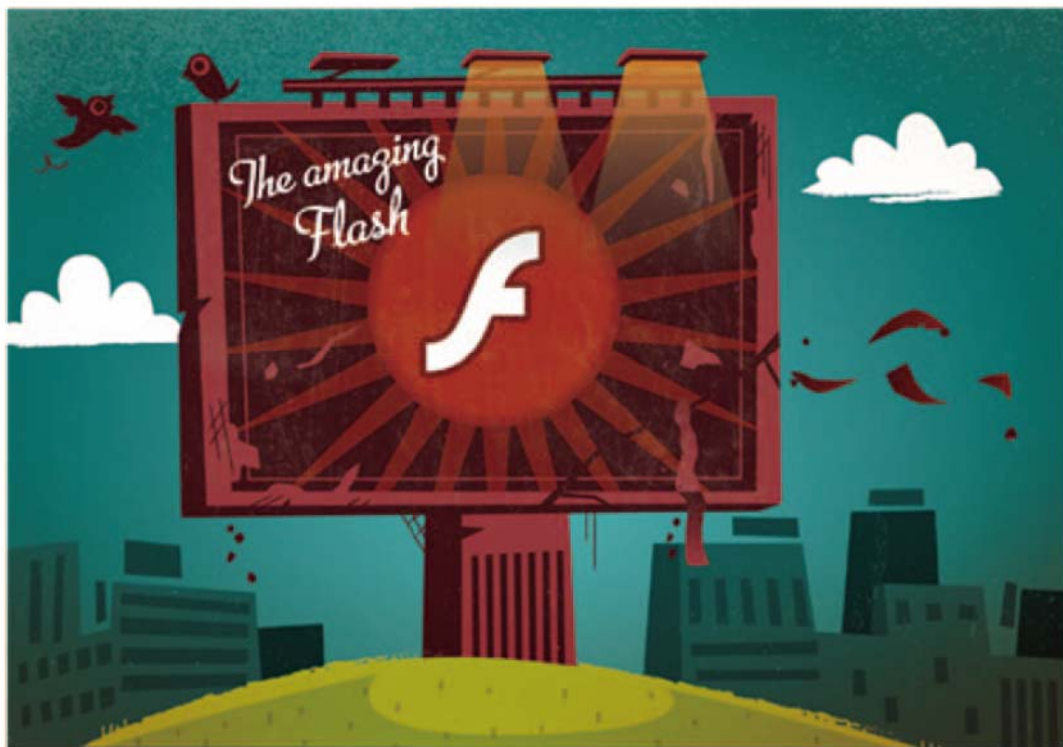
公司主页：[www.denachina.com](http://www.denachina.com)

请以“职位 + 姓名 + CSDN”作为应聘邮件标题，将中英文或中日文简历发至 [hr@denachina.com](mailto:hr@denachina.com)

# Flash之死

文 / Francisco Kattan

Adobe公司决定停止面向移动浏览器的Flash开发工作，但导致Flash最终失败的原因却不甚明朗。VisionMobile特邀作者Francisco Kattan撰文就导致Flash消亡的一连串事件进行了分析。



一颗Adobe多年前埋下的定时炸弹导致了Flash的消亡

自Adobe宣布将停止开发面向移动浏览器的Flash以来，人们就展开了激烈讨论，很多Flash开发人员表示惊诧和怀疑，分析师们不停地放马后炮，甚至有人恳请Adobe辞退首席执行官。Flash是一个引发人们强烈情绪反应的主题，其中既有充满激情的开发社区，也有歇斯底里的批判主义者。虽然我本人是Adobe拥护者，也曾在Adobe任职，但我会暂时搁置感情，本着坦诚的态度，基于公开的信息，客观地总结问题根源。

我认为，既不是HTML5扼杀了Flash，也不是Steve Jobs扼杀了Flash，而是Adobe多年前“不经意间”植入的定时炸弹导致Flash的消亡。

虽然Adobe未能迅速适应iPhone的各种调整，导致了面向移动服务的Flash消亡，但Adobe早已埋下了导致产品失败的种子。为了了解问题的根源，首先让我们回顾一下iPhone上市前的市场状况，以及这些事件怎样为后来的事情埋下伏笔。

## Flash Lite 时代

早在2000年中期，手机制造商的需求量巨大，他们乐意花钱购买Flash Lite（当时Flash的移动版本），因此Adobe决定对每台设备收取软件许可证费用。正是这一决策最终导致移动Flash的消亡。



亡，同时也将促使桌面Flash的消亡，这一点将稍后在本文中进行阐释。Adobe想要创建平台提供丰富的Internet体验的壮志未酬。

在许多人的心目中，都有这样一个强烈疑问，为什么Adobe并未复制PDF和桌面Flash Player所采用的成功模式套路：产品免费运行，并通过不断提高的工具收入获得利润。或许这将促使Adobe优先实现平台一致性而不是尽可能广但是分裂地覆盖各种平台。但事实并非那么简单。

虽然日本的Flash Lite生态系统曾经是一片欣欣向荣的景象（开发人员创建内容并通过经营者进行分销），但最初其他国家/地区并不使用Flash Lite作为应用程序平台。在很多情况下，手机制造商希望为核心应用程序（主屏幕、拨号器、通讯录、消息发送和通话记录等）构建富有表现力的用户界面，促使自产设备脱颖而出，他们才会采用Flash Lite。

LG Prada就是一个能够使用Flash Lite构建这种用户界面的绝佳实例。这种设备的特色在于拥有类似iPhone的触摸界面。Samsung D900和LG Chocolate也都是很好的例子。虽然这些设备包含Flash Lite，但它们并未为开发人员提供机会分发基于Flash的内容。因为浏览器中没有安装Flash，也无法执行基于Flash的应用程序，所以Flash Lite对第三方开发人员是封闭的。由于开发人员没法开发，也就没有工具收入，Adobe就只能从手机制造商中收取设备费用了。

## 冲突对象：手机制造商与开发人员

后来，即使出现了Flash Lite作为应用程序平台的机遇（尤其是Nokia和Sony Ericsson设备），Adobe也无法立即调整自身商业模式。回想起来，这错误损失惨重。那时，手机制造商希望自己的设备脱颖而出，而开发人员需要跨设备使用一致平台，他们的需求从根本上是冲突的。由于手机制造商买单，且移动团队根据收入衡量成效，因此Adobe优先满足手机制造商的要求，而将开发人员的要求置于次要地位，它允许手机制造商实现Flash以满足自身的各种需求。手机制造商获得Adobe授予的许可源代码，并创建自己的二进制

实现方法，而这些实现方法在各设备之间并不一致。Flash Lite有时用于构建设备用户界面，有时用于浏览Flash内容，而有时则用于运行独立的应用程序。此外，OEM并不总是实现同一组API，从而为开发人员带来了其他的设备分裂问题。更糟的是，由于运行库无法通过无线网络进行更新，因而设备问题只会随时间推移而变得越发严重。

## 开发人员缺乏分发和赢利机会

即便将Flash Lite作为平台进行部署，以便运行独立的应用程序，开发人员也无法方便地发布自己的应用程序。当时尚未出现iPhone风格的App Store。开发人员不得不通过中间商（集成服务商）发布自己的内容，集成服务商抽成，并与手机制造商和网络运营商之间订立了发行协议。

更糟的是，制造商和运营商并不具备良好的销售渠道，至少，消费者发现应用程序概率非常低。Flash开发人员无法方便有效地接触消费者。这是开发人员面临的主要问题，也是Adobe的主要隐患。与此同时，手机制造商收入的持续增加（支持Flash的设备出货量每年至少翻一倍）掩盖了问题的严重性，从而促使这颗定时炸弹的爆炸期限不断逼近。

## 一线生机：与运营商展开合作，深入消费者群体

为了给开发人员创造一个欣欣向荣的生态系统，Adobe将目光转向运营商。在当时，与运营商展开合作并不是一个明智的举措。对于那些刚刚接触移动领域，对运营商提供的控制水平感到不满的Adobe Web开发人员来说，更是如此。Adobe曾与多个运营商开展合作，但最突出的应该是与Verizon Wireless之间的合作。理论上，Verizon Wireless是一个理想的合作伙伴。Verizon Wireless是全球最大的CDMA运营商之一，对手机制造商具有重大的影响力，并且能够在其设备上指定Flash的运行库。当时，Verizon Wireless在美国还设有最成功的应用程序商店（BREW GetItNow）。

Adobe和Verizon联合推出了两项服务：一项是Flash应用程序下载服务，另一项是Verizon“Dashboard”，这是基于Flash Cast移动设备门户的更有野心的服务。但两项服务均存在问题。由于开发人员很难推出新的应用程序且所占收入份额过低，消费者很难发现应用程序，同时Verizon认证Flash新手机的速度过慢，所以BREW Get it Now服务最后以失败而告终。

Dashboard服务失败则是由于其推出时间过长，错失了最佳市场时机。Verizon于2007年3月对外公布Dashboard服务，并宣称将在下半年推出该项服务，但直至2008年9月才推出该项服务。即便那时，该项服务也只有Verizon商店强大设备阵容中的一种手机使用。随着iPhone和Android设备开始吸引所有开发人员的关注，Flash Cast和Dashboard才引起人们注意，但为时已晚。

但值得一提的是，Flash Cast和Verizon Dashboard的相关创新具有相当广阔的前景。这项服务与iPhone的许多关键属性如出一辙：同iPhone一样，它也具有“应用程序商店”概念，以便用户在其中搜索和购买新产品，同时也为开发人员提供一定的收入；同iPhone一样，它也是封闭的，并且有运营商监管；同iPhone一样，因为这些新产品和用户界面均基于Adobe Flash，所以它也以富有表现力的用户体验为特色。

不过，与苹果不同的是，Adobe并未对生态系统进行端到端控制，因此一再推迟该服务的上市时间。该项服务专为2006年的市场而设计，2007年iPhone的面世彻底改变了市场规则。虽然Adobe也在快速进行创新，但由于其依赖行动迟缓的合作伙伴，所以其创新思路无法及时传达给客户。

## 乔布斯和iPhone时代

iPhone的推出戏剧性地颠覆了移动生态系统，它以当时人们尚不甚明了的方式打乱了所有企业的思维。

这种影响非常深远，它对于依据这些新规则从零起步的新进成员（苹果和Google）更为有利，而非当时已经确立市场地位及拥有完善的商业模式的企业（Nokia、RIM、Motorola、Palm、

Microsoft、Qualcomm/BREW、Symbian、Sony Ericsson，当然还有Adobe）。与其他企业一样，Adobe无法快速地适应这种变化，因而付出了巨大代价。现在让我们来看一下产业生态环境发生的三大主要变化，以及它们对于Adobe Flash造成的消极影响：苹果导致当时的运营商一败涂地，而此时Adobe正在全力与这些运营商构建生态系统。消费者开始背弃功能手机，转而购买智能手机，但Adobe此前一直专注出货量更大的功能手机。移动浏览服务最终成为主流服务，但Adobe的移动播放器全都不支持Steve Jobs要求的桌面Flash内容。

第一代iPhone并不具备应用程序商店或SDK。iPhone是第一款具有卓越浏览体验的手机，并迅速占领了大量移动浏览市场份额（即使它当时的设备出货量所占的市场份额非常小）。在当时，缺失Flash功能是一个巨大败笔。

如果说乔布斯需要Flash，那么一定是在2007年第一代iPhone面市的时候。不幸的是，Flash当时并未准备就绪。由于Adobe从出产设备中赚取收入，因而它一直以来关注的是代表更大成交市场份额的功能手机（在Web浏览器页面浏览方面的投入几乎为零）。Flash Player的任何一个版本都不能满足乔布斯的需求。因为Flash Lite一直针对由更多限制的设备进行优化，所以不支持网络上的所有Flash内容；与此同时，由于智能手机要求具备桌面计算机的强大功能，因此无法在智能手机上有效运行所有的Flash Player功能。乔布斯曾说过一句至理名言：“这就是中间产品缺失导致的悲剧”，指的正是这一问题。

令人难以置信的是，直至2010年6月Adobe才推出完整的移动版Flash Player，而此时iPhone已经面世整整三年之久！

当时，iPhone已经成为全球最受欢迎的手机，苹果也已将重点由Internet浏览转移到与Flash毫无关系的应用程序上。Adobe已经错过了融入iPhone的大好机会。2010年苹果仍可以采用Flash平台，但在当时这显然不是其最佳选择。最终，由于Flash将会限制苹果实现其设备的差异化优势，因此苹果决定放弃Flash平台。苹果的营销策略侧重为能够在iOS上实现最佳运行效果的应用程序



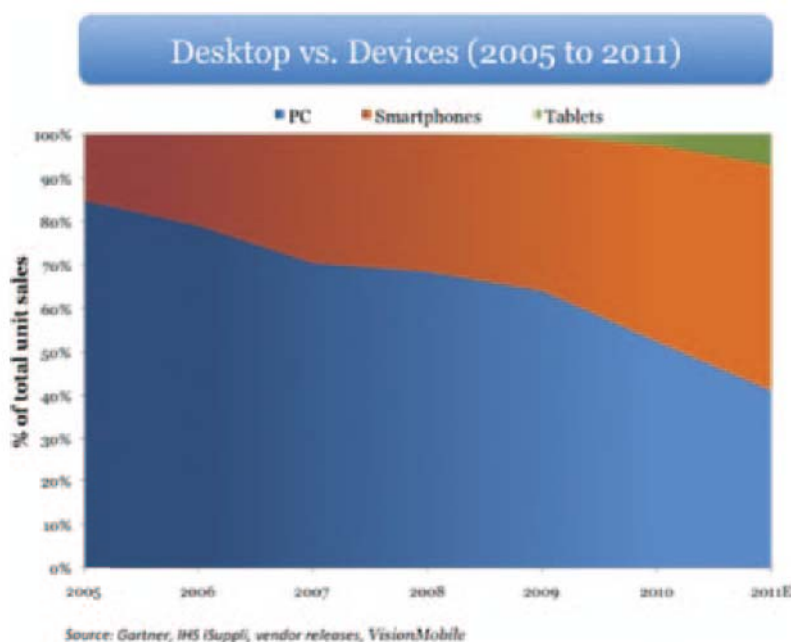
效力。为了支持这种定位，苹果需要开发人员以一系列最新的专有API为目标，而不是以写入更高级的跨设备平台为目标。这就是苹果决定从iOS中驱逐Flash的原因。

Adobe确实对iPhone弃用事件做出了积极反应，但可惜为时已晚。2008年3月，Adobe推出了开放屏幕计划，本质上而言，只要手机制造商要求开发人员以一致的方式实现，Adobe就免费为手机制造商提供播放器。为了确保开发人员保持一致性，Adobe也借用Windows和Mac OS在台式机上的一贯做法，开始为各种领先的移动平台创建自己的二进制播放器实现方法。然而，随着“无Flash”的iOS设备逐渐引领市场以及移动设备和门户网站越来越多地采用HTML 5，形势已经明朗化，Adobe已无力回天。

面向移动服务的Flash退出历史舞台，而面向桌面服务的Flash却将继续存活，是这样吗？大错特错！道理非常简单：由于缺乏移动支持，面向桌面服务的Flash依旧无法生存。随着PC在Internet连接设备中所占的份额越来越小，大部分网站也将逐渐进行更新，从而不再需要Flash，这只是时间问题。举例来说，许多门户网站会为了采用Flash而拒绝Internet的大部分具有吸引力的特色内容，这一点令人难以想象。

当然，网站没有必要彻底删除Flash内容。他们可以添加逻辑，以便支持面向桌面的Flash内容以及面向其他设备的HTML5内容。事实上，多年内，这将成为向“无Flash”Internet过渡的方式。当摒除Flash的新内容横空出世，HTML5的普及率和功能赶上Flash，且PC所占的市场份额继续下降时，Internet上支持Flash内容的网站比例将趋近于零，从而导致桌面Flash渐渐消亡。

请注意，这种过渡已于数年前悄然开始，各门户网站纷纷自我调整，以便支持iOS设备，而iOS设备现已占据移动网页浏览份额的62%。YouTube也开始支持HTML5，这对Flash造成了很大打击。jQuery这个一直在构建交互网站方面与Flash竞争的JavaScript库已经超越Flash。HTML5浪潮正在起势，桌面Flash遭受与其移动同胞相同的厄运已成事实，这只是时间问题而已。



PC的市场份额正逐渐减少

## 总结

很多年前，Adobe在就已埋下在Flash领域的失败恶果，即当时具有重要意义的收入模式，但长久以来一直是一颗尚未引爆的定时炸弹。这一模式导致Adobe迷失方向，朝着背向市场最终趋势的方向发展，特别是在iPhone发布之后（功能手机与智能手机、手机制造商需求与开发人员需求、利用运营商作为渠道与利用苹果和Google作为渠道）。此外，当推出iPhone后，Adobe适应新型市场现实的步伐过慢，历经3年才发布Flash Player 10.1，最终导致Flash毁灭。🔴

作者简介：Francisco Kattan在移动生态系统领域工作长达10年之久，曾任Adobe移动业务部产品营销和开发合作总监。他还在Edify和Openwave担任过领导职位，如今在Alcatel Lucent担任产品管理高级总监一职。

感谢www.visionmobile.com授权，如需引用，请与《程序员》联系。

# 逆势：衰退的开发者信心和不断示好的开发环境

文 / 郑金条

目前，惨淡的收入似乎难以支撑开发者继续走下去的信心，然而本文作者却指出，开发环境正在改善。

我在《程序员》2011年6月刊提到了一个现象：不管是个人开发者还是规模化的团队，都希望自己赶上移动互联网大爆发的时代。诸如《二战风云》（顽石互动）、《捕鱼达人》（触控科技）、《Gamebox》（Trinity）的成功利好信息；《Angry Birds》（Rovio）整体超过6亿次下载的超级数据；智能手机季度出货量的大规模攀升，以及类似苹果App Store月度应用下载超过10亿次规模和年度超过20亿美元的开发者分成，让整个市场更加信心爆棚。所有的运势都明显指向着：成功和荣耀期待被复制再复制，以至于每一个参与者都能从这场硝烟角逐中凯旋。

## 急转直下的评判

而最近，不管是社交游戏还是手机游戏，消极的声音反倒此起彼伏。在经历了一段时间的艰涩探索和碰壁之后，看衰获益前景的心态反倒趋于兴盛。这可能和几大因素紧密相关。

**其一，应用的曝光规则。**在曝光频度最高的苹果App Store，同样因为排名规则而使新的App很难有更好的机会冲破既有的App排名格局，开发者除了要兼顾好游戏的质量本身，如何实现更好的营销渠道是同样困难的一件事情。

**其二，低门槛和山寨心态的不良后果。**开发者群落中的先作恶再从良心态比比皆是（比如先试水市场行情，比如先累积用户，比如有限的研发经费）。但市场看起来正在反向阻击类似情况，各

种超级游戏的移植和大制作的出现，提高了一般用户对手机游戏的认知，不再是简单休闲的小游戏，也可能是一些大资本投资的力作，无形中使山寨的门槛被拔高（比如制作成本大幅度攀升）；此外，业界对山寨的抵制也使山寨开发者很难获得与成就相应的认同地位（山寨将是我们行业发展路上的一杯毒酒，只有饮鸩止渴的暂时满足感）。

**其三，来自资本的支持和现金流压力增大。**和电子商务目前所遇到的情况相似，不能获得投资的开发者依靠自身的资本无法支撑收支平衡。在经历一段时间的投入之后，现在对于开发者最困难的部分，无疑就是来自资金方面的压力，这种压力将改变企业对产品的态度，同时也将改变企业成员的心态。

**其四，团队的心态。**对于开发者来说，除了流动资本的困难，另外一个更大层面的打击是内部成员的信念。面对市场开拓的乏力和企业运营的压力，随之而来的就是成员的动摇。这种动摇包括对企业的未来信心，也包括对自己付出不得回报的失望。

## 成本、格局和竞争

Ben Britten曾撰文指出开发者初期的游戏无法获利是正常现象，认为这些要么是蹩脚的尝试，要么是熟悉和了解行业以及发行平台或者媒介环境投石问路的测试品。但对于开发者来说，前期游



戏研发周期不宜太长，要么周转资金不允许，要么团队锐气在没有经验的前提下被消磨殆尽，到最后完成开发任务就是胜利，更谈不上修缮。

事实上，每个开发者所面临的都是绝对残酷的竞争环境，很难有足够的姿态去应对长期的不盈利格局。在社交游戏领域，Rick Thompson认为他们新用户的广告获取成本已达到2~3美元（非自然获取或者病毒式渠道获取），而Kixeye更是认为他们的用户成本是4美元（Hardcore领域）。尽管手机游戏的成本没这么夸张，但按照Fiksu历次追踪的数据走势可以清晰地看到在苹果App Store忠实用户（最低三次打开同一个应用）的获取成本已从3月的0.94美元上升到9月的1.64美元（以0.99美元一次有效下载算，定价0.99美元的这个下载成本已经相当高昂了）。如果换算成人民币，单个有效用户的获取成本就在10元以上的高水准了（难怪现在很多应用都转向免费下载，但依靠高昂的IAP来获得营收）。尽管也有人说他们不一定要消耗这么大成本的广告费用去获得一个有效用户，可以在平时或者节假日时通过限免来吸引庞大的用户群。但我们可以看一个残酷的数据，在日常情况下，苹果App Store每天都有700款左右的应用在限免，遇上光棍节（1500款以上）或者Black Friday（1400款以上）这种活动更是铺天盖地，其实竞争这种曝光机会的难度一点都不亚于寻常的营销难度。

并且这种竞争还有加剧的趋势，3月Getjar所做的一次调查显示，用户获得App信息的渠道仍相当分散，按照Getjar的数据，应用商店对App曝光的贡献仍然逊色于在线网站，但在VisionMobile的最新调查中，应用商店的曝光贡献已经超过45%，远远高于其他的形式。换句话说，外围营销的式微，就意味着在应用商店内的曝光机会争夺将更加激烈，成本的持续高涨在所难免。

另一个明显的竞争来自庞大的App数据。Mobilewalla分析认为App总数将近100万，并且每天以新增2000款的速度快速增长，尽管苹果App Store总下载量超过了180亿次，而Android Market的总下载量突破了100亿次，但是大量应用的下载量低于100次是不争的事实。既定格局的力量几乎根深蒂固，在苹果App Store和Google的Android

Market中Pocket Gems、Zynga、Storm8、Rovio、Game Insight、Backflip Studios、Funzio、TinyCo或者The Playforge长期把控着营收榜的分布（美国为例），而新进入者又面临着重重的曝光障碍，种种矛盾正在累积，以至于大量的开发者在市场壁垒面前看不到继续前行的新希望。

顽石互动的吴刚在微博上分享了他们突破各种屏障的方式，不管是《契约OL》上了Symbian还是《二战风云》上了WAP或者iOS平台都是寻找到平台的产品空档而找到了存活并且获益的机会。

我们曾解析过为什么同样在付出，有些开发者能够寻找到生存的方式，有些反而轻易被市场淘汰，或许有一些症结是值得思考的：比如看起来没什么差别的画面、雷同的玩法，相似的营销策略甚至相同的布局渠道，理不清自己游戏的核心卖点，逐步在市场的大同小异中失去光芒。

其实至今为止还是有开发者对自己所处的格局，如何导向自己的成功或者导致自己的失败充满困惑，只能参透机遇在背后巨大的推动价值。

---

仅有7%的iOS应用实现收支平衡，更是只有3%的开发者实现了盈利，但并不会太多的开发者觉得这个市场暂时失去了机会，反而有更多志在一搏的豪情。

---

Rovio因为《Angry Birds》声名鹊起，但两年之后用户只能继续期待着Rovio重复再对《Angry Birds》进行多次包装，将一招走遍江湖进行到底，但这背后潜在的巨大成功就是两年来收获5亿以上的下载量和3000万以上的日活跃量。Chair Entertainment旗下的《Infinity Blade》起初只是为了测试Unreal Engine 3在iOS中的应用，最终却成为收益超过2000万美元的超级游戏，而其后续力作《Infinity Blade 2》更是横扫多个国家和地区的付费下载榜和营收榜，甚至还带出了这一模式的更多游戏选项，诸如Glu Mobile的《Blood And Glory》、Warner Bros的《Batman Arkham City Lockdown》，并且后两者在营收方面都有强劲的表现。而Zynga在手机游戏端的各种大手笔收购和各种社交游戏的手机版移植，造就了新的手机

游戏巨人。按照Zynga的说法，现在他们在移动端的DAU已经达到了1300万的高度（Storm8在6月时才400万的DAU）。

## 以App Store为例的机会

手机游戏的市场可能正在趋向于好转，我尝试着从五点进行论述（以苹果App Store为例）。

**其一，市场机会。**IHS iSuppli的研究数据认为2011年苹果App Store营收总额为29.1亿美元，如果按照70%的分成比例测算，那么开发者总共能够分成20.3亿美元（并且北美iPhone用户家庭收入10万美元以上的占47%，远高于Facebook用户的11%，能够从消费力上做更多的保障）。如果要从市场细分的话，可以看到和中国开发者利益更休戚相关的三个市场：美国市场，按照Mixpanel的说法，2011年美国手机游戏市场的产值可以达到15.3亿美元；日本市场，Yano Research数据称2011年日本手机社交游戏市场为14亿美元（Seed Planning的数据估算更大一些为21.79亿美元）；韩国市场，最大的利好消息就是韩国游戏评级委员会对苹果App Store和Google Android Market的游戏类别进行解禁，在产值方面，按照Korea Creative Content Agency的说法今年预计为4.5亿美元。

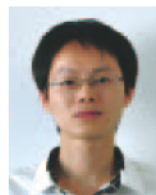
**其二，苹果App Store更好的曝光机制。**早先苹果App Store的曝光环节包括10个付费下载排行、10个免费下载排行、10个营收排行、14个新应用推荐、14个热门推荐、14个职员推荐等86个曝光位置（这个领先于Android Market的65个）；现在苹果再度对推荐功能进行优化，不再从下载层面考量而是从游戏属性层面再细分，比如Innovative Gameplay、Intuitive Controls、Stunning Graphics、Quick Fixes以及Hardcore，真正能够对应玩家的游戏类型偏好，也能够给到不同的游戏类型以更多的曝光机会。

**其三，人民币支付问题。**尽管人民币支付可能在外界看来充满了潜在的政策风险，诸如手机支付的牌照问题、文化部颁发的文网文（文化部网络文化许可证）、新闻出版总署的游戏版号（以及游戏备案）、苹果App Store的ICP经

营许可证以及上线游戏的游戏著作权等各种未知的问题。但因为实现了本地化支付使各种非境外支付账号也可以流畅消费，再加上0.99美元和6元人民币之间的结汇差异（更便宜），数据说明了问题：据Distimo的数据称自从App Store实行人民币支付以后付费下载前100名的下载量几乎翻了一倍。从现在这个层面上看，人民币支付暂时给出的是利好消息。

**其四，手机游戏的市场走势被看好。**最典型的数据来自Flurry对美国portable games市场长达两年的对比追踪。Flurry数据显示在这个市场环境中iOS和Android游戏从2009年的19%份额递增到目前的58%，而Nintendo DS则相应地从70%下滑到36%。

**其五，iPad系列被认为是新一代掌机的替代选择。**Spazzmania首席执行官Lin Tam认为iOS已可以制作出运行更快且更有深度的游戏；Ogmento联合创始人Brian Selzer称新设备已经可以帮助开发者实现更具真实感的游戏。事实上，情况也正往这些方向发展。Chair Entertainment旗下的《Infinity Blade II》大小为939MB；Gameloft旗下的《Chaos Rings for iPad》大小为777MB；inXile Entertainment旗下的《The Bard's Tale》大小为1.34GB，并且在营收排行榜上相当有竞争力，在12月2日的Top Paid榜单上有第一位的《Infinity Blade II》（6.99美元）、第四位的《Call of Duty: Black Ops Zombies》（6.99美元）和第十位的《The Bard is Tale》（5.99美元）。我们还可以从Localytics的数据分析中得到更多论证，他们认为iPad的两大功能分别是阅读和游戏，Giancarlo Mori (Glu) 也称iPad游戏更趋向于HardCore玩家。P



郑金条

游戏邦负责人，游戏邦主要关注和解析国内外社交游戏和手机游戏领域，并定期做深度行业阐述。

责任编辑：陈博（chenbo@csdn.net）



# 《丁香园用药助手》成长之路

记者 / 陈博

在移动化的道路上，丁香园团队遇到了哪些挑战与抉择？《程序员》将请他们分享面向垂直专业领域应用的成长之路。

丁香园CTO冯大辉曾透露自己是因为身体不好才萌生做健康产品的念头而离开支付宝的。接受《程序员》采访时，冯大辉说这一半实情一半是玩笑话。他选择加入丁香园最主要的是认同丁香园做的事情，认同丁香园团队。2011年，丁香园团队也紧跟移动互联网浪潮，推出了移动应用——《丁香园用药助手》，这款在7月推出的产品，发布5个月，下载量约为75万，在App Store排名最高上升到中国区免费榜第三名。

## 移动化是大势所趋

丁香园一直是以话题为主的医学论坛，网站会员大部分是医生。丁香园CEO张进认为，在移动互联网时代浪潮袭来之时，作为一个以关系为基础的产品，丁香园必须拥有各种移动互联网设备的客户端。这些产品将提升丁香园的医生社会化媒体属性，为医生的交流带来更大的便利。

产品需求大多来自其目标用户的反馈。丁香园曾做过一个非常简陋的Web版用药查询平台，出乎意料的是，这个工具的用户却非常多，很多用户打电话来希望丁香园能够提供移动终端的产品。在做过用户调查以后，丁香园认为这是一个非常好的切入医疗移动互联网的机会。所以在2011年3月，丁香园技术团队提出了移动端应用的设计方案，希望推动这个产品来迈入这个领域。于是，丁香园专门成立了6人的移动团队，在产品经理李宁的带领下开始尝试移动应用的开发，这也是他们第一次在移动中做的尝试。产品从3月份开始设计，8月份全面上线。

在一开始，李宁他们主要希望通过这款产品解决医生在实际工作应用中的问题。比如，他们发现

平时医生在用药时，需要处方，或者药典，或者借鉴国外已经成熟的产品，所以他们想在这个方面做尝试，希望为医生提供一个更便捷的软件。于是《丁香园用药助手》的计划就应运而生了。

李宁说，由于国内很缺乏类似应用的开发经验，所以在早些时候，丁香园技术团队成员也走访了美国一些专注于手机医疗应用的创新上市公司，对国外医药行业的移动应用做了一些分析，他们的成功经验给了团队不少启示。但因为中美国情和医药政策的差异，药物信息在各个国家都有其专有的特殊性，美国市场上医疗相关的优秀软件未必适合中国用户。所以丁香园将出发点定为中国医生临床诊治提供用药参考工具，其主要特点还是在医疗方面的专业性——只有立足专业性，才会为用户创造价值。

因为一开始就有明确定位，所以丁香园网站的用户就成了《丁香园用药助手》的第一批用户。2011年6月份软件第一版上线，团队就邀请了网站的专业医生安装使用。应用的功能都是针对医生的需求进行设计的，也直接给用户带去了方便。随之而来的就是不断地口碑传播，用户量也不断得到增长。据统计，丁香园的用户中有40%~50%的用户是医生，这些用户基本上都是丁香园的会员。

## 路遇挑战

有了清晰的定位和思路，并不意味着一帆风顺。冯大辉表示，作为一款面对医生群体的应用，在开发用药助手时，丁香园技术团队主要需面对两个方面的挑战。

第一个挑战是数据的准确性：因为用药助手的使用对象有很大一部分都是医生，使用场景是临床



丁香园移动开发团队

诊治过程，因此数据必须准确无误，而国内很少有这类用户行为的调研数据。因此他们成立了数据处理团队，并且在数据源上做了非常严格的把控，只采用经过SFDA批准的正式临床药品说明书中的数据，由数据处理团队进行编辑，摒弃了一些“药典”中的脏数据，做到实用、准确。

第二个挑战是针对医生这个特定的用户群的App交互设计：与IT人士和极客相比，医生可能会对通用的设计元素感到陌生，也不愿意花时间来学习应用如何使用。另外，在临床诊治过程这样的使用场景之下，用户需要用非常少的步骤在非常短的时间内找到需要的信息。而在产品研发初期，很多用户的使用习惯都只能按照行业传统的方式去揣摩。为了解决这些问题，应用交互设计和前端工程师做了大量用户研究和测试。

环境障碍

与大多数国内开发者一样，《丁香园用药助手》的成长路上，不仅仅存在自身的问题，整个环境同样为丁香园的前行之路设置了很多障碍。

在李宁眼里，智能移动终端让个人健康管理的数据采集和监控变得快捷方便，给慢性疾病健康管理和突发事件的医疗救助带来很多便利。比如血压监控，设备根据便携和通用，从测量到数字化存储的时间缩短了一半，接入互联网后，信息可被医疗机构或健康管理机构事实获取和监控。可

目前，虽然软件和硬件的开发，在国内已经不存在技术障碍，但国内还没有能够提供较好健康管理的机构，因此即便是个人的健康数据信息可及时采集，但无人监控、挖掘和服务，之前的数据采集也就失去了意义。

另外，虽然近些年来WiFi已经迅速普及，但国内很多用户依然在用速度较慢的GPRS网络，甚至有不少地区的营运商对用户访问的网络进行了限制。

李宁说，在产品上线之初多少有一种试水的心态。但在第一个版本发布之后，整个开发团队收到了来自各方面的积极反馈，正是这些用户反馈让他们坚定了对产品的信心。

前路漫漫

虽然目前产品取得的成绩不错，但李宁坦言，用药助手在各方面都还不够完美，目前最急需改进的问题是其产品的迭代节奏仍不够快。移动应用在功能和使用习惯上都跟Web应用差别很大，把控开发周期和迭代速度十分关键。并且，作为一款以数据为核心的搜索类应用，如何在确保数据全面准确的情况下，让用户既能实现本地搜索，又能尽量控制软件体积大小，也是团队当前正在着手解决的问题之一。

他们发现，在医院有很多患者会拿着正在吃的药片去看病，这个场景让他们想到了按外观找药品的功能。因为很多药品在外观表面都会有一些特点，他们就从颜色、形状、剂型等几个关键点入手，对药品属性进行了标注，进而实现了图片搜功能。但同时，大量数据的整理耗时耗力，给数据支持团队带来了很大挑战。

《丁香园用药助手》提供了普通版和专业版，普通版涵盖了大部分常用药品，能基本满足药物信息查询的需要。专业版包括了更多药品信息、专业用药指南、医学常用计算工具。专业版用户还可以发送免费短信对患者进行用药指导。而在张进眼里，《丁香园用药助手》只是迈入移动领域的第一步，团队正在为刚刚推出的《丁香客》研发相对应的客户端，他希望丁香园基于以上应用，延展到医患互动的社交应用。P



## “建立企业中的用户体验力量”系列之五

# 用户体验专业团队的建设思路

文 / 吴卓浩

用户体验团队的建设包括其组织形式、人员构成、成员招聘、业绩考核、团队管理等一系列学问。

### 用户体验团队的组织形式

用户体验团队的组织形式在功能上，有执行型、研究型、服务型、驱动型、综合型之分。大多数企业选择的是执行型和服务型的用户体验团队，用户体验团队通常在产品设计基本确定以后才进入研发，主要从事细化和实施产品设计的工作，这种组织和工作形式比较容易控制，对用户体验团队成员能力的要求也相对较低。单纯的研究型或驱动型用户体验团队通常只在研究机构或者设计公司中出现，帮助客户做产品设计的前期探索，但不会深入到产品研发的迭代和实施中。用户体验工作只有全程融入研发过程，才能发挥出最大贡献，一个具有综合功能、并且能够带动整个企业开展用户体验工作的用户体验专业团队才是最好的选择。

在企业中，用户体验团队的组织形式有独立型、附属型两种。它们各有优缺点。独立的用户体验团队利于企业中跨团队的用户体验资源积累与协调，以及用户体验人员之间互相激励与成长；但这种组织形式对工作管理、绩效评估带来一定难度，如果用户体验人员不能保证在一个项目团队中的积累，往往会遇到领域知识和团队协作方面的挑战。附属型用户体验团队的优点正好对应独立型团队的缺点，其缺点正好对应独立型团队的优点。这两种组织形式分别适应不同规模的企业或者企业的不同发展阶段。一定要注意扬长避短，独立用户体验团队需要努力培养用户体验人员在具体项目中的积累和执行力，附属型用户体验团队需要努力做好跨团队、跨产品的思考和资源流动。

用户体验团队所能发挥的能量与团队规模没有必然联系，主要与团队的成员质量以及在企业中的影响力有关。大企业的用户体验团队规模虽然大，但因为项目众多，具体落在一个项目上的用户体验人员数量反而可能较少。另外，不同类型的企业、企业在不同发展阶段所需要的用户体验人员比例也不相同，企业越小、越早期，所需要的用户体验人员比例越高，因为在创造产品最关键的起始阶段，用户体验工作可以发挥出最大的价值。在谷歌、微软、腾讯、百度、阿里，用户体验人员与工程师的人数比例在1: 20~1: 150之间，而在创新工场，我们投资的创业公司中用户体验人员与工程师的人数比例平均大约是1: 5。

### 用户体验团队的人员构成

以下是有关用户体验职能名称的不完全列表：

用户体验设计师 (User Experience Designer, UX Designer, UE Designer, User Experience Architect, User Experience Specialist)

产品设计师 (Product Designer)

信息架构师 (Information Architect)

交互设计师 (Interaction Designer, IX Designer)

界面设计师 (User Interface Designer, UI Designer, Interface Designer)

图形界面设计师 (Graphic User Interface Designer, GUI Designer)

视觉设计师 (图形设计师, Visual Designer, Graphic Designer)

游戏美术 (Game Artist)

可用性工程师 (User Experience Engineer, UE

Engineer, Usability Engineer, Usability Specialist)  
 人因工程师 (Human Factor Engineer, Human Factor Specialist)  
 用户体验研究员 (User Experience Researcher, UE Researcher, UX Researcher, User Researcher, Usability Researcher)  
 人因研究员 (Human Factor Researcher, Ethnographic Researcher)  
 前端工程师 (Front-end Engineer, Front-end Developer)  
 界面工程师 (User Interface Engineer, UI Engineer)  
 原型工程师 (Prototype Developer)

通常来说,你可以选择以下几个基础类型的人员来构成你的团队。

**用户体验设计师:** 综合能力强,侧重产品和交互设计,兼具视觉设计、原型制作、用户研究能力。

**交互设计师:** 侧重产品和交互设计,了解视觉设计、原型制作、用户研究。

**视觉设计师:** 侧重视觉设计,了解交互设计、原型制作、用户研究。

**界面(前端)工程师:** 侧重前端编程,了解交互设计或视觉设计。

**用户体验研究员:** 侧重用户研究、实验设计、数据分析。

这几类人员的数量配比取决于产品特质、企业(团队)的发展状态,并没有一定之规。



## 用户体验人员的招聘

用户体验人员是一群特殊的人。相比于工程师,他们更游走在感性和理性、理想和现实之间。他们的工作是不停地和不同的开发者合作、争执,一面跟自己较真,一面跟合作伙伴较真。要想让优秀设计师与你同行,一定要有共同的梦想。双方都要明白,你们找的不是一个替代品,不是一个劳动力,而是携手实现梦想的人。

面试考察设计师和考察工程师在思路是一样的:专业技能是基础,但更重要的是分析能力。面试前预先查看面试者的作品和简历,基础设计能力通常能够从作品中看出来。在面试中间开放性的问题,刨根问底,深入了解细节、分辨真伪。如果不能从回答中确定面试者的能力,就现场出题让面试者动手设计。除了考察专业能力,还需要考察面试者与他人协作、影响他人的能力,因为设计师的工作中大量的与产品经理、工程师、其他设计师的协作。对于有的资深人员或者有培养潜力的人员,还可以考察领导力,看面试者是否具有在企业中推动用户体验工作的能力。

许多有优秀用户体验团队的企业招聘信息本身就是很好的参考材料,从中你可以了解他们对用户体验人员的要求以及背后的考察目标。以下用创新工场的用户体验设计师招聘信息为例来说明。

### 用户体验设计师——职责描述

■ 配合产品经理、工程师,进行产品的概念原型设计及细化的交互设计,配合进行用户测试及分析。解读:要从产品概念设计阶段就加入。概念设计需要多方面的支持,包括交互设计、用户研究,以及对市场、相关/竞争产品的理解。

■ 对新产品或已有产品进行界面设计并配合工程师实现,注重细节,提高产品用户体验。解读:用户体验设计师要设计新产品,也要改进已有产品,并要负责直到设计在产品中实现。

■ 通过建立用户模型、故事板、竞品分析、头脑风暴、流程图、界面/产品原型、设计规范等方法,对设计创意进行有效的交流。解读:前两条都强调和不同角色的人的合作,有效的交流想法在合



作中尤为重要，要掌握和选择合适的方法。

■ 能在快节奏、跨平台的环境中，迅速学习、同时支持多个项目。解读：这样的环境对自我学习、管理的要求很高，有类似经验的人能更迅速适应。

### 用户体验设计师——职位要求

■ 对用户界面设计相关工作有浓厚兴趣，学习能力强、富于创新精神。解读：没有要求学校专业，而是强调“兴趣”、“学习能力”和“创新”。怎样展现自己在这些方面的能力和潜力？

■ 具有良好的沟通协调能力，富于团队精神。解读：在学校、生活中都有很多锻炼团队能力的机会，你在其中做过什么？

■ 具有出色的设计表达能力，能迅速有效地将想法表现为设计方案。解读：根据不同的情况，有时要快速而有大局观，有时要准确而注重细节，有时要发散而富于创造力。你不妨尝试对同一个题目做完全不同的设计表达，来锻炼自己。

■ 熟悉互联网、移动互联网产品的研发流程和方法，有实际工作经验者优先。解读：有机会在企业实习将会给你补上这一课。

■ 视觉设计出色者优先。前端编程出色者优先。解读：用户体验设计师是领域交叉的结果，侧重在交互设计和产品分析；但设计表达和实施的能力，如视觉设计、前端编程，会在产品设计、工作协作中给你带来很大的帮助。

### 用户体验人员的业绩考核

以企业内部的用户体验团队为例，可以主要从下面几个方面评估用户体验人员的工作。

■ 完成设计或研究的质量和效率。衡量专业能力，可以参考用户体验评估的方法，并结合工作合作伙伴和用户的反馈。

■ 推进和沟通设计或研究的态度与方法。衡量协作能力，只是提出好的解决方案不行，还要高效的推进方案的实施。

■ 提高用户体验在企业中的认知和意识。通过讲座、工作坊，建立和梳理规范，收集整理设计资

源和模板，建议改进工作流程等，推进整个企业的用户体验工作。

■ 推动产品改进和创新。跨产品工作的用户体验人员更有机会比专一的产品经理、工程师发现不同产品之间相关联的问题和机会，应鼓励由此推动产品改进和创新。

另外，用户体验人员常常跨产品、跨团队工作，在考核上需要综合多方反馈，就他们所参加的项目给予不遗漏、不过分，公平公正的奖励。

### 用户体验团队的管理

以下是我的四条心得体会，抛砖引玉。

1. 关心团队成员的需求和梦想，帮助他们争取资源、制订计划来实现，奖励他们取得的成绩。每个人都不会甘于平庸，但再高远的理想也要以现实中一步步的积累和努力为基础，一句简单的鼓励、一个看似小小的机会、一些坚持和督促，可能会改变一个人的一生。让希望纵向、横向发展的设计师都能获得充足的空间。

2. 激励团队成员，加强交流，让团队有健康的风气、有持续的成长。一个人想三天和三个人想一天的效果绝对不同，设计是最需要思维碰撞的职业之一。用户体验人员的个人成长非常依赖于团队氛围，团队和个人都需要持续性的成长。工作的空间是否利于协作，工作的方式是否鼓励协作，团队的机制是否促进协作，都有实事可做。

3. 帮助用户体验人员以更大的视野看待产品以及和其他团队沟通协作。与其他团队的沟通协作不仅是技巧问题，更是视野问题。当设计师执著纠结在一个设计点上时，需要思考这一点在全局中的位置、对于其他团队的轻重缓急，调整自己的工作。经常跨功能、跨产品工作的用户体验人员有机会看到产品的全貌，应当鼓励设计师对于产品全局乃至战略的思考。

4. 努力提高自身素质，在专业上做到服众，和团队成员一起战斗。最好的领导会雇用比自己更好的人才，但在技术团队，在专业上服众永远是重要的。好领导的价值在于成就团队成员的价值和成长、补全团队的忽视和缺失、思考战略和强化

执行,让每个人和团队都更出色。如果在专业上不精通,就无法洞察问题,更提不出建设性的解决办法。

再深入一个层面,用户体验团队的管理也兼具在整个企业中推动用户体验工作的目的。

■ **明确用户体验团队在企业中的定位。**设计在IT企业中的定位,往往出于一个现实:工程师可以把好想法变为产品,设计师需要工程师把好想法变为产品;设计团队无法独自成功,而是帮助其他团队一起成功。正确的认识到用户体验团队的定位,努力做好本职工作,再以此为基础,探索和驱动新的机会,才能真正扩大在企业中的影响力,提升用户体验工作的地位。

■ **将用户体验为中心的思想植入企业的核心价值和企业文化。**用户体验工作无法由用户体验人员单独完成、而是由整个企业中不同职能的人员共同完成。用户体验工作首先是一种思维方式,只要以正确的方式思考,就可以有效地参与用户体验工作。让不同职能的人员都认识到,产品需要以最好的用户体验状态出现在用户手中,并以此为共同的目标,调整工作的方法和流程,让用户体验思维真正融入团队的各项工作中,才能做出真正具有优秀用户体验的产品。

■ **充分组织和利用资源让用户体验工作产生结果。**再好的想法,如果不实现出来也只是空想,就不会发挥影响力。用户体验团队不仅要高质量高效率地完成平时工作,也要主动去组织力量啃一些硬骨头、打一些大仗,利用用户体验独特的视角发现问题和机会,并且调动开发、运营、市场等力量,实现想法,帮助企业在产品甚至战略上取得一些突破性的进展。这就是在整个企业中开展用户体验工作的独特价值。

## 用户体验团队的业绩考核

用户体验团队首先是一个业务部门,和技术、运营团队类似,都要有相应的业务考核标准,通常也要有团队之间的互评。但用户体验团队还有一些特殊之处,需要额外的考核机制。

用户体验团队不仅是用户体验工作的主要承担

者,也必须成为企业中提高用户体验意识、普及用户体验知识的推动者。发起讲座、工作坊,分享用户体验资源、案例,让更多不同职能的人员具备基本的用户体验设计与研究知识与技能。建立设计规范、资源和设计模板,推荐设计工具、做使用培训,让产品经理、工程师和其他设计师能够更高效地开展用户体验工作。建立定期的产品用户体验评审机制,组织用户测试和数据研究,让产品的用户体验质量控制体系逐步完善。

用户体验团队对企业的贡献可以不仅仅局限在产品的用户体验工作上,还可以拓展到改进“企业”本身这个产品上——留心观察、采访一下身边的同事,你会发现大家在平时的工作中有许多不尽满意、有所期许的事情,有些可能是小事,比如前台的访客管理,食堂的客流引导,会议室的投影设置;有些则不见得是小事,比如怎样的办公桌椅搭配最有利于员工的健康,怎样的会议室设置能满足大小会议的需求并且利用率最高,有什么工作流程或方法有问题、如何改进。这些改进和你的产品没有直接的联系,但却是高效、愉快工作的基础,会在潜移默化中影响你的企业和产品。

优秀的用户体验团队和具有优秀用户体验的产品更是一个企业的名片。用户体验团队以其独特的视角和思维方式,可以从产品、运营、营销、品牌、招聘等方方面面为企业找到独特而又高效的运作方法。一些公司的优秀用户体验团队在网上开设博客和微博,讨论、分享用户体验的知识和经验,这有利于建立积极的企业形象,也有利于整个行业的进步。

用户体验团队不是一个一般性的技术团队,而是企业中跨团队、跨职能的用户体验工作的重要执行者和推动者。其组织形式、人员培养、团队管理和绩效考评都需要结合企业自身的状态做因地制宜的选择,并从推进整个企业中的用户体验工作的角度来考量。P



吴卓浩

创新工场用户体验总监,曾创建谷歌在中国的用户体验团队。

责任编辑:陈博(chenbo@csdn.net)



# 移动应用开发技术选择六要素

文 / 李亮

作者从平台环境、操作系统、设备能力、云端、应用类型、跨平台开发六大方面分享了其在移动应用开发中的技术选择经验。

这是一个新的时代、新的机会。自从2007年1月乔布斯揭开iPhone的面纱以来，移动时代的大潮滚滚向前，已经走过近5个年头。这个产业正在从新生走向成熟阶段。在这样的产业时代背景下，各种不同的系统平台，不同的技术路线，自然是层出不穷、迅猛发展。对于每一个投身于移动行业的软件开发者而言，充分地了解和掌握各个平台的发展状态，搞清楚各种开发技术与方向的优劣，是关系到产品与自身发展的头等大事。

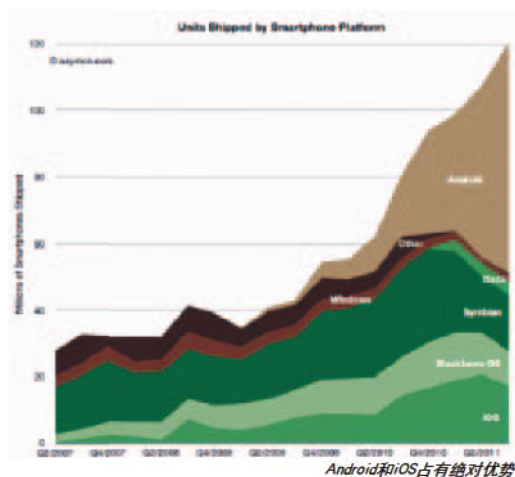
## 平台选择

从2007年到现在，涌现出了不少面向移动设备的操作系统平台。先不论原先已存在的Symbian和Windows Mobile系统，新的系统就有苹果的iOS、Google的Android、Palm/HP的WebOS、Nokia的Maemo/MeeGo、三星的bada，以及RIM新的基于QNX的PlayBook Tablet OS。

一个真正占有市场的平台才能吸引更多开发者为其开发更多应用，更多应用又能反过来替平台争取更多用户，从而促进平台的进一步发展。正是由于平台与应用之间相辅相成的关系，使得平台的选择成为开发者首要关注的重点。

对于开发者而言，一个移动平台的意义，并不只是一个操作系统而已，它还包括了与之相联系的整个生态环境。平台的市场占有率直接决定了应用能够被多少消费者使用，平台本身又能带给开发者多少回报。这些都是在平台选择中必须考虑到的问题。

无疑，iOS和Android是目前占有市场份额最大的两个平台。这两个平台不但用户众多，拥有的开发者数量、应用数量和下载量也都远远高于其他平台。一些数据可以反映这种情况：在今年最新的北美地区智能手机市场份额调查中，Android及其定制版占有了58.67%的市场份额，iPhone占有14.67%，Symbian占有14.33%，BlackBerry占有9.2%，bada占有3.33%，而Windows Phone仅占有1.67%的份额【数据来源：<http://www.asymco.com/2011/11/17/the-global-smartphone-market-landscape/>】。



作为一个平台，iOS有以下几点特点：第一，iOS是一个相当一致的系统。目前苹果仅有三种运行iOS的设备：iPhone、iPad和Apple TV。从API方面来说，这三种设备的产品各自的设备属性都相当固定，而各自的操作系统版本也都保持一致，这对开发者来说是一件极其方便的事情；

第二，iOS系统本身具备了相当良好的，内建流畅的用户交互实现。就算是一个照着教科书中中规中矩写出来的iOS程序，其用户体验也会不会差；第三，苹果提供了一个非常好的iOS程序开发环境。只有很少的功能是模拟器无法模拟而必须在真机上进行调试测试的；最后，苹果本身的品牌也是一个相当有影响力的支持。

另一个主要的平台就是Android。得益于Google对Android的开放（不向厂商收取授权费），以及大大小小各种厂商的机海战术，Android手机的占有量已经是第一位了。因此，选择了Android平台就意味着选择了最大的用户群体。另外，Android本身源码的开放性，对于一些需要利用底层实现细节的开发者来说，是个很好的特性。而且从开发本身来说，Android所采用的开发语言和环境，相对来说比iOS的门槛要低，这是它的优势所在。但也正是因为Android的开放，造成了市场上Android设备的规格特性各不相同，增加了应用中机型适配所带来的大量开发和测试工作。

除了iOS和Android这两大巨头之外，Microsoft的Windows Phone 7从去年开始也大张旗鼓地开始发力。在WP7平台上，Microsoft仍然扮演着系统提供商的角色，硬件生产仍然是各个厂商的事。但Microsoft为了保证用户体验的一致性，对设备的指标作了非常严格的规定。因此，目前WP7还没有机型适配的问题。在新的移动市场上，Microsoft显然来晚了一步，成为占有比较

小的市场份额的竞争者，这一点远远比不上iOS和Android。另外，WP7提供了专有的Metro UI界面形式，展现了跟iOS和Android都截然不同的风貌，这对其是有益的。从开发本身来说，Microsoft为WP7开发者提供的工具有比较大的限制。除了Silverlight和XNA之外就没有别的支持了。很多非Web Based的、非游戏类应用无法在现有的平台上进行开发。这不能不说是一个弱点。

三者之外，除了BlackBerry在北美地区的占有率较高以外，别的平台都只占有少少的一点份额。BlackBerry的系统，已是一个落后整个产业十余年的东西，BlackBerry赖以生存的基础是集成到系统中的push mail机制，而不是它的系统本身是先进还是落后。Symbian则是连Nokia自己都放弃的东西，不会对现在的开发者有多大的吸引力。webOS生不逢时，空有良好的用户体验和很有特色的开发方式，却最终被HP所抛弃。三星的bada则仿佛是其自身的一亩三分地，只在wave系列中推出。

从市场占有率来看，开发者首选的平台无疑是iOS和Android了。除了已占有的市场份额，以及目前所能看到的发展前景之外，对开发者来说，还有一点非常重要：这个平台对开发者的回报如何，对于开发者的支持如何。在这一点上目前苹果比Google做得好。各种调查统计数据表明，在iOS平台上的开发者收入要高于Android平台。另外，苹果对从设备到应用的严格控制虽然为人诟病，但也不可非议地为应用开发者创建了一个单一而良好的环境。反过来，Google的开放政策变成了不管不问政策，由于没有审查机制，导致Android Market里有很多质量低劣的应用，甚至是恶意应用。官方的Android Market尚且如此，各种第三方的商店市场就更不用说了。

当然，在做平台选择时，可能还会有一些其他因素存在。比如在iOS上，应用无法读取用户的短信记录，也不能在不通知用户的情况下发送短信或者拨打电话。对于有这样需求的应用（我们姑且先不谈是否恶意或者侵犯隐私）就只能在Android上开发了。因此，对于平台的选择，市场占有率、生态环境的优劣、以及平台是否能提供所需要的具体功是几个比较重要的考虑因素。



平台之争仍在继续，期待新平台的崛起



## 技术路线的选择

### 操作系统

首先考虑的是操作系统。操作系统是开发应用的基础，操作系统的能力提供也限制了开发的应用程序的能力。其实，在选择平台的部分，我们已考虑到了操作系统。因为现在的平台本身，就是以设备，运行在其上的操作系统，以及围绕着应用开发和用户建立起来的一个生态环境组成的。在设备厂商的设备差异努力被操作系统隐藏起来，运营商慢慢沦为纯粹的数据带宽提供商的今天，我们通常都已经直接把操作系统的名称用作平台的名称了。不过，在选择平台时，我们会更多地考虑到宏观因素，而做具体的技术选择时，就要多考虑一些具体的技术因素。

**系统是否支持多任务。**支持多任务意味着可以编写运行在后台的服务程序，可以在不干扰或者不通知用户的情况下做一些工作。目前，iOS系统只支持一些有限的多任务，而Android则对此不加限制。所以，如果有很强烈的多任务需求的应用，Android会是一个更好的选择。

**操作系统本身对应用能做的事情有什么样的限制。**相对来说，iOS对应用的限制较大，应用程序运行在一个单独的沙盒中，不能读写任何其他应用的数据，只能访问很少一部分系统允许应用访问的数据。而Android限制较少，绝大部分系统数据都能访问。

**安全性和隐私。**应用能访问的系统数据越多，所能获得的用户信息也就越多，用户的隐私和数据安全就越难以得到保障。从这个角度看，iOS 显然更优势。

**使用同一系统的不同设备之间的差异。**这是一个跟操作系统和具体设备都相关的重要因素。差异越大，应用开发中需要对不同设备作不同处理的地方就越多。iOS一致的设备指标使开发者不用在这方面花什么心思。反观Android，设备的差异非常大，从屏幕分辨率大小到有无键盘，从电阻屏还是电容屏到有没有浮点处理器，每一款都有其自身特色和不同之处。这样就使得应用开发者需要对大部分不同的设备都作测试才能发布到市

场上，很多时候还需要针对不同的设备编写不同代码。说得再远一些，那些基于Android开放的源代码修改出来的系统，往往会给开发者造成很大的困扰。

### 设备能力

在技术路线选择中，设备本身的能力也是重要因素。设备本身的能力包括了设备自身的计算能力，即CPU的性能，以及是否具有浮点处理器等；图形能力，是否具有图形处理器加速、GPU是否支持OpenGL、支持哪个版本等。这些功能对于计算密集型和图形密集型的应用，比如3D游戏，是极端重要的。从更细节的角度来讲，CPU支持什么样的指令集，对于比如视频解码这样的应用来说，都是十分重要的因素。设备本身的能力还包括交互部分，即屏幕大小、分辨率、是否具有硬键盘等。屏幕大小直接决定了屏幕上能显示能操作内容的多少。一个为手机设计的UI在平板上会显得很丑陋，而一个为Galaxy Note设计的UI在QVGA上根本都没法看。而硬键盘的存在与否，不但跟系统本身相关，跟应用程序本身也相关。在一个有硬键盘的手机上，就有更多空间来显示内容。而Android下的输入法，对硬键盘的输入处理跟屏幕软键盘的处理就完全不同。移动设备上所附带的各种传感器或数据采集设备，也是跟应用的技术路线息息相关的。目前很多应用都依赖于GPS、摄像头这些附件。

### 云端

云端，即服务器端。在这个移动+互联的时代里，完全脱离云端的应用越来越少，而社交网络的兴起加上移动对社交的天然亲和性，使移动应用对服务器端的依赖度更高了。从跟服务器端的交互来看，需要考虑的技术有四个方面。首先，需要云端提供什么样的服务。这是最基本的一点，无须多说。其次，移动端以什么样的方式从云端取得数据并数据发送到云端。是采用通用的协议还是私有协议，数据是否需要加密等问题。再次，移动端与云端之间的通信需要多少带宽。是否在现有的2G/3G网络条件下能够得到满足，抑或必须使用Wi-Fi。移动端与云端之间的通信是必须

长时间保持还是可以断续进行。移动应用的网络环境会比桌面复杂得多,网络带宽会比较快而频繁,并且会经常在有网络连接与无网络连接的情况下转换。最后,还需要考虑一个问题是自己搭建服务端还是直接利用现有的公开服务。自己搭建服务端能够更好地与自己的移动应用进行配合,能够自行定义把哪些计算放到服务端进行,但需要在服务端的软硬件上投入与维护。而利用现有的服务则比较省事,代价是移动端需要做更多的工作,有些功能会难以做到。



云服务在移动互联网时代尤为重要

### 原生应用还是Web应用

大部分面向互联网服务的移动应用都会面临这样一个问题:是开发一个移动设备上的原生应用,还是直接开发一个适应于移动设备上浏览器的Web应用?这两种形式各有利弊。原生应用的优势在于能够充分利用设备的能力,达到比较好的表现效果和更加平滑的用户体验。有一些能力是Web应用所无法做到的,比如需要用到加速度测量仪的应用,需要用到摄像头的应用,这些只有原生应用才能做到。另外,原生应用的好处在于即使没有网络连接,很多本地功能都可以继续使用,而在有网络连接的情况下,把一部分功能由原生应用来实现,还能大大节省流量,并提高应用程序的响应速度。

Web应用的优势在于,开发者不需要具备专门的移动平台开发知识,并且能够随时对问题进行修正和发布新版本,能够更快地让用户体验更新,符合目前流行的先发布再慢慢改的风气。这两者

中,一方的优势恰恰就是另一方的劣势。针对不同类型的应用程序以及开发人员的具体状况,需要有具体的判断。这两者的优缺点都同样明显,因此将长期共存下去,共同繁荣,不存在一方消灭另一方的可能。

### 跨平台

随着各种不同的系统出现,也出现了不少立足于跨平台的解决方案。本来在PC上最强大的跨平台方案Flash在被苹果拒之门外后,由于其本身在移动平台上性能和电池消耗方面的不足,最后退出了市场。但新兴的以HTML 5为口号的新一代跨平台方案纷纷出现,其中最著名的是PhoneGap。这类以HTML 5为旗帜的跨平台方案内容都很类似:开发者利用HTML 5/CSS/JavaScript编写应用,编写的应用经过本地的一个定制过的浏览器中间层运行。这些应用都还是原生应用而非Web应用,只不过以开发Web应用的形式进行开发而已。这样的框架,通过对本地浏览器框架的定制,将设备的特殊能力封装成能够被JavaScript访问的系统对象,从而能够充分利用设备的能力。这样的跨平台方案有很明显的优势:开发一套东西运行于各个移动平台,平台间的差异由框架解决;不需要学习各平台各自的语言和API,只需要会用HTML 5/CSS/JavaScript开发即可,降低了开发成本。

不过,我们也能很清楚地看到它的劣势:中间的运行适配层有相当大的运行开销,性能与原生代码相比有较大损失;因为需要带入定制的浏览器框架,最终生成的独立应用体积相对比较大,需要使用更多的资源。20世纪80年代以来,无数人有着跨平台梦想,从桌面到服务器到移动设备,都希望一套东西包打天下。可惜到现在这还只是美好的愿望而已。我个人认为,这世上没有一统江湖的可能,跨平台对于一些应用或者对于某些开发者来说是个利器,但它并不能成为统治者。P



李亮

程序员, iCosta for iPhone以及Rockplayer for Android的主力开发者之一。有20余年软件开发经验,从嵌入式系统到企业级开发均有研究。目前致力于移动计算与云计算方向。

责任编辑:陈博 (chenbo@csdn.net)



# Mobile Web App发展现状及展望

文 / 刘铁锋

作者指出，虽然性能上还存在很多缺陷，但从技术角度分析，Mobile Web App并没有太多致命门槛，正处于构建框架，重建规则的过程中。

在计算机的发展过程中，目前移动设备的时代可谓潜力最大、发展最迅猛也是竞争最为激烈的一个时代。硬件的发展速度令所有的消费者惊叹，如果是2008年，一部主频528MHz、内存192M、屏幕3.2寸、分辨率320×480的手机会让你感觉十分前卫，因为CPU和分辨率足够高。而放到3年之后的今天，这部机器已经沦落到无人问津，成为古董机型的地步。不错，这就是G1的配置。而现在双核1.5G的CPU、4寸甚至5寸的超大屏手机已开始成为用户心中的主流手机。毫无疑问，这种手机的处理能力已经开始能够赶上PC的处理能力。而Pad，更有赶超PC发展速度的势头。

在硬件处理能力不断强大的同时，移动设备的操作系统的竞争也开始越来越激烈，不仅是iOS、Android，还有不知道市场反响如何的Windows Phone。而仅是这三种平台，就已经开始让传统PC转向移动的开发以及企业开始头疼。与此同时，HTML5能力的不断强大，使得Web App和Native App之争不断升温，在移动设备上的讨论尤为突出。但无论如何，当硬件能力的进一步强大，性能问题得到改善之后，快速开发程序的能力以及构建程序的成本会成为影响和推动技术选择的一个重要原因。

## Mobile Web App的现在

作为一个坚定的Web App支持者，我认为，开发难度和跨平台的需求会在技术和商业两个方面都会推动Web App得到越来越广泛的应用。市场对Web App的接受程度也在不断地得到印证，大家

所熟知的Mobile Web App先行者Finance Time，在短短3个月时间内已经获得超过100万的下载量。相比起很多原生的程序，这个下载量不算太高，但其几乎匹敌原生应用的体验，已经让很多人留下了深刻的印象。

不仅如此，还有很多其他数据可以用来分析Web的趋势。如大家所熟知，Facebook也一直在推进Web的发展。Facebook不仅收购了HTML5的移动应用开发商Strobe，同时也正式推出了名为Spartan的HTML5移动开发平台，其目标就在于更好地帮助开发者开发Web App。不仅如此，最近有消息传出，作为在Web App开发的最火的工具/框架之一Sencha的技术推广经理跳槽去了Facebook，足见得Facebook正在不断积蓄力量以求更大的突破以及发展。而另一则Adobe放弃在移动设备上支持Flash的消息，更让人看到了HTML5在同一技术标准以及能力上的突破。

与此同时，国内对Web App的技术的关注也非常



Web App将得到更广泛的应用

活跃。一个些力于探讨和分析业界对Web App的最新进展和发展趋势的博客已经出现，并且质量相当之高。不仅如此，PhoneGap中文站也已经浮出水面，为国内的开发者带来了全中文的教程以及学习资料。

这一切都在说明业界对Web App不仅仅停留在口号上，而是有更多的实质性的推动。

## Web App开发现状

某技术博客里面的一篇博文对Web App做了如下的定义：

“要给出完整的Web App的定义是一件很复杂的事情，因此我们在此只给出一个简单的定义：Web Application是指通过使用Web和Web浏览器技术，跨越网络完成一个或多个任务的应用程序，通常需要使用Web浏览器。”

简单的说，就是利用Web技术，能够做出超越传统理解网站的功能，让它更具有交互体验，让这个App看起来和用起来更像Native App。这样就非常清楚Web App和Native App的差别具体在哪里了。在目前的技术储备上Web App同样也开始用户Cache、Drag & Drop等Native App所必备的功能。

那么从开发层面来看，Web App开发和Native App开发又有怎样的差距呢？

让我们回想一下Native App的开发过程。

■ **界面开发。**一般来说，Native App的界面开发拥有非常强大的控件库。不管是对用户交互的Button、Checkbox，还是用户输入的textbox、RichTextbox，或是用户展示的ListView、GridView之类的控件（不同的开发平台下控件的名称未必一致），控件库里已经为这些控件的展现方式，为属性设置、事件响应等基本的需求做好了充分准备。对于绝大部分的开发者，只需要拖拽控件，就可以实现自己的逻辑代码，而无需做太多的准备工作。

■ **事件响应/数据绑定。**在绝大部分的场合下，对于数据的处理成为Native App开发中的重要工作。这部分工作的本质就是把从网络上传的一

组数据（不管是从数据库，还是从Web Services）转换成为业务逻辑中所定义的对象，然后绑定到相应的数据控件中。而事件响应的过程则是相反的过程，根据用户的响应，修改相应数据控件里面的值或者状态，然后通过数据处理逻辑回传到数据库或者Web Services中。在这个过程中一般来说Native App的开发过程中，同样有逻辑处理非常完善的库来帮助实现这个工作。比如说Android中的Content Provider或者Adapter。

■ **数据状态管理。**数据的状态管理是指根据实际的开发需求所带来的本地缓存，配置文件读取等操作。比如聊天应用、文本编辑应用或者基本的信息管理系统都有能力直接从本地的磁盘中读取之前的操作记录或者缓存信息，从而让程序能够有能力很快地启动并且展示，例如聊天程序中的联系人列表/邮件客户端里的本地邮件等。这些都依赖于有本地的存储和缓存，来让用户更快地获取信息。

这里没有强调具体和网络操作/具体和业务相关的逻辑处理等具体需求。对比在Web App开发的过程来说，情况则不太一样。由于传统的Web展示能力有限，传统的Web开发过程中，由于位于前端。因此界面的开发本质上让位于了CSS所创造的效果以及JavaScript所包装出来了各种各样的交互效果。这部分的工作更多地集中在样式的调整以及和动画效果制作上。

对于所谓的事件响应、数据绑定等方面，在早期的开发过程中基本上不存在这个概念，完全跳转回服务器，然后重新刷新页面。在AJAX引入之后，利用JavaScript + XMLHttpRequest，使得HTML的页面逻辑可以转移到JavaScript中实现和完成。

对于数据的状态管理部分。本质上来说，早前的Web基本不存在这个概念，仅有的Cookie能力有限，仅能部分保存状态。直到Web Storage开始实现，甚至是Web Database的出现，才增强了Web在这个方面的编程能力。

那么，回过头来看Web App和Native App之争的本质是什么？无非就是Web App是否能够完整实现Native App所能做的事情。这样Web App在开



发简单/跨平台方面的能力才能充分凸显出来。但对现阶段的情况来说，当期望把Native App所拥有的功能转向到Web App时，不可避免地因为工具的缺失，甚至是开发理念的缺失，导致了在开发中始终存在不够的状况。可以坦率地说，现在Web App的开发中，还处于构建不同的开发工具和开发库，甚至是在摸索开发模式的过程中，还尚未成熟。

当然，这不是不能解决的问题。Web App开发，尤其是Mobile Web App的很多工具和框架已经开始组建建立。

■ **界面开发。**如前面所说，非常强的Native App界面控件成为提高效率的有效保证。在Mobile Web App开发中，Sencha以及jQTtouch已经提供了非常强大的界面开发支持。同时在界面库方面，jQuery Mobile可以认为是一种增强型的就JavaScript库，能够有效地帮助用户来解决和提升开发效率。

■ **事件响应/数据绑定。**和Native App相比，这块本来就不是Web开发的强项，但也许根据开发的需要，未来会衍生出包装得非常完善的一站式解决方案。比如可以直接把一组RSS里面的内容，更加方便地变成具有交互能力的List或者功能。

■ **数据逻辑/缓存处理。**Web Storage已经提供的技术的支持，需要的就是最佳的开发实践，甚至是利用缓存的模式。对于简化开发的工作来说，应该会出现专门的Storage的管理模式，甚至是封装得非常完善的库。

## 现有Web App开发模式的问题以及挑战

■ **性能问题。**性能问题依旧是非常大的挑战。由于Web App的开发几乎完全构建在Webview的基础上，因此在Webview上对事件的处理以及响应的能力就直接决定了用户的体验。在这里有两方面的性能，一个性能是对事件响应的速度，在Web上控件的响应速度比原生的控件响应速度要慢；另一个是直接在渲染和执行速度上面的速度。

■ **分辨率的问题和适配的问题。**和Native App的开发方式一样，不同的分辨率、横竖屏切换，以及

对于不同机型的识别，甚至与对不同的Web浏览器内核的适配，同样存在一样的问题，同样需要比较多的调试和适配的工作。

■ **离线的问题。**和Native App相比，可能这是最应突破的一件事情。一是这是一个0或者1的问题，实质上是突破了原有的Web开发的限制，二是界面和逻辑数据的分离。对于Web的页面来说，这可以认为是界面，中间涉及分离的JS、CSS文件以及没有更新的img等静态元素的缓存问题，同时也存在把动态数据元素（比如某个控件里的状态、离线邮件中的邮件信息）缓存以及载入的问题。这需要重新建立起一套解决方案来实现。

■ **跨平台问题。**Web App和Native App另一个不同在于访问硬件资源上的不同。由于受限于浏览器的功能，有很多的硬件资源不能直接访问。所幸是PhoneGap为在多个平台上开发提供了非常的好解决方案，使得Web App拥有了能够在多个平台上执行的能力。有理由相信这个问题能够被解决得越来越顺畅。

## 总结

从开发的工具以及各种支持来说，Mobile Web App开发尚处于比较早期的原始阶段。但从技术角度来看，并没有太多致命门槛，而是处于构建框架、重建规则的过程中。不仅是Facebook、Google这样的互联网巨头，连IBM这样的老牌公司也已经在开始搭建Mobile上的工具以及解决方案，这让开发者完全有理由相信，Mobile Web App的开发会越来越火热，让我们拭目以待。P



刘铁锋

百纳信息技术有限公司CTO，W3C标准化组织成员。带领团队开发的海豚浏览器在CNET评选的2011年全球最受用户欢迎Android应用中排名第二。

责任编辑：陈博（chenbo@csdn.net）

# 寻找机遇 创造未来

## CSDN 热门职位全新推荐

### 方正宽带网络服务股份有限公司

**FOUNDER 方正**

正在你身边

#### 公司诚聘:

- 高级研发工程师 (Linux C方向)
- 高级研发工程师(Java方向)
- 高级研发工程师(PHP方向)
- 软件测试工程师

#### 任职要求:

- 本科及以上学历, 计算机或网络相关专业
- 熟悉相应的开发语言, 有相关开发经验者优先
- 良好的团队合作精神, 较强的创新精神

发送简历请以“姓名+职位+CSDN”为标题

邮箱: hr01@founderbn.com

公司网址: <http://www.founderbn.com>

### 友邦创新资讯科技

友邦创新资讯科技是友邦保险集团中国香港总部之分支机构, 宗旨是为友邦保险集团属下亚太地区的公司提供应用软件开发和资讯服务。公司拥有完善的培训计划, 并有海外培训机会。



#### 现诚聘如下职位:

- Java Engineer/Analyst
- ASP engineer(strong DB skill)
- Cognos/DataStage Developer
- COBOL Programmer for Insurance System
- Tester/Junior BA

我们热诚欢迎你加入本公司, 简历接收邮箱: HR.BJ@aia.com

公司地址: 东城区东单北大街

### 杭州拓乐网络科技有限公司

#### 诚聘高级网游程序员若干:

- 逻辑程序员
- 客户端程序员
- 服务器程序员
- 网络工程师和计算机管理员

更多岗位信息请访问公司网站: [www.tolez.com](http://www.tolez.com)



简历投递: hr@tolez.com 联系人: 徐小姐

### 上海易睦

imo是国内基于互联网的企业级即时通讯平台。也是“全免费、注册即用”服务模式的开创者与引领者。



#### 岗位:

- IM服务器架构师
- IM服务器开发经理
- Web架构师
- Web研发组长
- 即时通讯客户端开发经理
- 即时通讯客户端高级软件工程师
- Android/iPhone手机软件工程师
- 高级测试经理
- UI高级软件工程师

**福利:** 高端VC+福利租房+项目奖金+带薪年假+期权+高薪+基本不加班+调休

简历投递: hr@imoffice.com

### 聚胜万合信息技术(上海)有限公司

聚胜万合MediaV依托强大的互联网广告平台, 为广告主、网站主和战略合作伙伴提供专业的互联网营销服务。



现热招云计算工程师、PHP工程师、数据处理应用工程师、产品经理等职位, 邀您共谋中国数字营销大变局。

如果您想在互联网广告领域一展鸿图, 如果您拥有足够的经验和才能, MediaV期待您的加入。

公司网址: <http://www.media.v.com>

### 赛门铁克

**公司简介:** 赛门铁克是提供安全、存储和系统管理解决方案领域的全球佼佼者, 可以帮助企业保护和用户保护并管理信息。



**部门简介:** SEP (Symantec Endpoint Security) 系列产品是赛门铁克面向企业级安全战略的核心。

#### 热招职位:

- 项目经理
- 软件测试工程师
- 软件工程师
- 实验室系统工程师

联系邮箱: DL-ENG-EMG-China-Recruit@symantec.com



## 完美世界（北京）网络技术有限公司

完美世界诚聘英才，欢迎与我们一起开启一个完美世界



招聘职位:

- 游戏引擎开发工程师
- Java开发工程师
- Flash (AS3)开发工程师
- PHP开发工程师

详细JD请登录<http://hr.wanmei.com/>

简历投递邮箱: [duzhipeng@pwr.com](mailto:duzhipeng@pwr.com)

QQ咨询: 2393130592 欢迎大家投递简历

## 天津港保税区科技发展局

东软（天津）、展讯（天津）高薪诚聘IT英才

- 高级Java工程师
- 高级C++工程师
- DSP工程师
- 芯片驱动工程师
- 应用软件工程师
- 驱动软件工程师
- 算法工程师

工作地点:

天津空港经济区 ([www.tjftz.gov.cn](http://www.tjftz.gov.cn))

简历投递: [resume@adm.tjftz.gov.cn](mailto:resume@adm.tjftz.gov.cn)

## 北京法国电信研发中心

Software engineer in the web service system development

Position requirements:

- Master degree in telecommunication, Compute Science OR equivalent
- Experience in J2EE, SOA, Spring, Hibernate, Struts, XML, REST, HTML5 development
- Experience in cloud computing development, such as SaaS, PaaS, etc.
- Strong programming skills, at least 3+ years experience in Java software development
- Familiar with the software development methodology
- Familiar with networking technology
- Good English communication skill
- Good team working spirit and communication skills, and hard working as well



Contact: [hr@orange-ftgroup.com](mailto:hr@orange-ftgroup.com)

## 广州瀚信通信科技股份有限公司

招聘职位:



- C/C++ 初级开发工程师
- C/C++ 中级开发工程师
- C/C++ 高级开发工程师
- C/C++ 系统架构师
- C# .NET初级开发工程师
- C# .NET中级开发工程师
- C# .NET高级开发工程师
- C# .NET系统架构师
- 研发项目经理 (C/C++ 或 .NET、Delphi等均可)
- 软件测试工程师
- 交互设计师
- 交互设计师助理

详细信息请登录CSDN网站查询! 请以“职位+姓名+CSDN”作为邮件标题, 将简历发送至: [yfhr@hantele.com](mailto:yfhr@hantele.com)

## 无锡迈达博尔信息咨询有限公司

无锡软件园, 占地800亩, 建成55万平方米, 在建37万平方米, 入园企业超400家。  
网址<http://wnd.iparkhr.cn/>



招聘职位:

- 无锡软件园——对日软件工程师
- 无锡软件园——Java开发工程师
- Android开发工程师
- 测试工程师

简历投递: [iparkhr@126.co](mailto:iparkhr@126.co)

## 方正国际软件有限公司

方正国际医疗卫生业务是中国医疗信息系统解决方案的提供商, 经过16年的发展, 已成为成熟的数字化医院及区域医疗卫生网解决方案的提供商和服务商。



为适应业务的蓬勃发展, 现在急聘

Java中高级开发人员 Unix/C高级开发

投递邮箱: [bfi\\_zhaopin@founder.com](mailto:bfi_zhaopin@founder.com)

公司主页: [www.founderinternational.com](http://www.founderinternational.com)

公司地址: 北京市海淀区北四环西路52号方正国际大厦5层

# Hadoop不是万能的

## Hadoop in China发起人谈Hadoop现状

记者 / 董世晓

2011年，对Hadoop来说注定是载入史册的一年：Yahoo!与风险投资公司Benchmark Capital联合建立独立公司Hortonworks，接手Yahoo!开源软件平台Apache Hadoop；Hadoop海量数据分析平台Datameer、Karmasphere和Hadapt分别获得千万美元级别投资；一向对开源趋势不屑一顾的微软转投Hadoop怀抱；Hadoop之父Doug Cutting第一次莅临中国，布道Hadoop 0.23……

可以说，Hadoop生态系统呈现一派欣欣向荣的景象。近日，记者专门采访了Hadoop in China的两位发起人——Yahoo!北京全球研发中心高级经理韩轶平和中科院计算所副研究员查礼，请他们为大家诠释Hadoop的现状。

### Hadoop商机无限

2011年关于Hadoop的众多投资以及微软等公司转向Hadoop的事件，昭示了业内感受到了Hadoop中蕴藏的商机。

韩轶平认为：“互联网行业由于拥有深厚的技术实力，所以Hadoop得以率先在互联网行业得到了蓬勃发展。但这并不意味着其他行业便没有商机。因为不是所有企业都有能力使用Hadoop，但需求又实实在在地摆在那儿，需要很多的技术支持、定制化产品，这便会最终催生出一个产业，来做这些方面的服务。”

查礼则解释了传统IT公司对Hadoop的态度：“微软、IBM这些公司拥抱Hadoop，并不是说它们在向Hadoop倾斜，而是将Hadoop作为一个标准来支持，将Hadoop加入到自己的产品体系或者业务架构中，并进而巩固其在这个市场上的地位。”

通过支持Hadoop，老牌IT厂商得以用最小的代价加入大数据处理这个新兴市场，不失为明智之举。

### Hadoop技术热点

#### NoSQL

从2010年NoSQL的好处被认识到之后，好像一夜之间在IT业内便已街知巷闻。目前，NoSQL已经不再停留在学习概念和坐而论道的阶段，而是得到了广泛应用，HBase、MongoDB、Redis等深入人心。应该说Facebook在这中间起到了领头羊的作用，率先采用了一些开源NoSQL系统来支持其大数据应用，而国内互联网企业也看到了这个趋势，并积极跟进，不断通过进行深入的交流和实践。

#### 实时处理

据韩轶平介绍，现在包括Facebook和Yahoo!在内的越来越多的公司都在向“实时处理”这个方向发展，而且会有更多的公司来关注怎样真正地让数据处理和分析能够越来越实时，因为处理接近实时之后，会对业务产生深远的影响。例如对在线广告系统，如果处理时间从十分钟、一小时减少到一分钟，广告收益会有本质的区别。查礼认为，在解决了实时处理这个“快”的问题之后，“准”的问题就迫在眉睫了。而Hadoop/NoSQL就能解决“准”的问题，从而实现精准营销，达到高利润、低成本的目标。

此外，Namenode、Scalability（可扩展性）、HDFS Performance这些存在已久的问题，仍然是Hadoop的技术热点。

应用需求或者业务需求，说到底是要为企业赚钱。





两位Hadoop in China发起人畅谈Hadoop热点（左为韩轶平，右为查礼）

企业对经营模式、盈利模式的追求，最后导致业务追求，这种业务追求就反映到对系统的追求上，所以这些追求决定了有哪些问题要解决。例如，High Availability（高可靠性）对电商来说至关重要，如果在圣诞节宕机一小时，损失可想而知。

## Hadoop应用场景

可能是长久苦于大数据处理手段有限的缘故，随着Hadoop应用的不断拓展，使很多人陷入了对它的盲目崇拜中，认为它能解决一切问题。

对此，韩轶平说：“在我们研发中心的每次Hadoop训练班开学时，我都会告诫学员务必记住‘Hadoop不是万能的’。在做具体工作时，首先要考量它是否适合使用Hadoop。”这正如Doug Cutting所说：“如果一个小任务用一个节点就能完成，为何还多此一举用Hadoop呢？”但如果任务当前很小，考虑到以后会有暴涨的话，最好还是一开始就用Hadoop来实现，以便保持将来良好的可扩展性。因此，在动手做之前，这些问题都应该要想清楚，如果没有把握的话，可以多参加一些社区讨论和交流活动，看看哪些人、哪些公司在做类似的事情、有哪些经验教训，参考过后，再去做选型：技术到底怎样构架、应用怎样来组合

这些技术。总之，做到不迷信、不盲从。

从狭义上来说，Hadoop就是MapReduce+HDFS，也就是一种离线处理或者批处理的模式。一方面，如果需求是对在线请求马上得到结果，Hadoop肯定不适合；另一方面，MapReduce是一个简化的并行模型，假设硬件总是要坏的，所以它中间所有的交互模式都做得很可靠、简单。这样一来，Hadoop能够处理的负载或者应用，就会相对较窄。

而互联网应用模式是一种简单模式，传统IT行业则复杂得多，路数与互联网企业完全不同，正因如此，Hadoop率先在互联网企业获得成功。但以后Hadoop极可能应用到传统IT行业，比如MIS或者ERP。在传统IT行业中，Hadoop是否保持原样，继续是专用系统，还是慢慢演化成为通用系统，现在尚未可知，但已经显露出这样的苗头。因此，查礼断言，Hadoop就像早期的操作系统一样，刚推出时是个专用系统，经过多年的发展，才成为通用系统。这也符合计算机行业中从专用到通用的螺旋上升趋势。

虽然目前Hadoop在大企业，特别是互联网大企业中应用较多，但韩轶平认为这并不是因为存在技术门槛的缘故，而是一方面要看企业是否需要做Hadoop，另一方面则是因为像Facebook、Google、

Yahoo!这样的公司得到了较多的媒体曝光，为人所熟知，而更多的小公司，不管是国内还是国外，都在用Hadoop，可能规模较小、业务相对简单。而数据的膨胀，也让我们很难定义何为“小”公司了：硅谷一家创业公司，员工只有13人，但在Amazon租用了1400台机器跑它的一个搜索业务。

## Hadoop主力谁属

Doug Cutting曾说：“Apache软件基金会的初衷是不希望始终有企业在某一个项目里面一家独大，我们希望有更多的人参与进来，希望有更多的交流和碰撞。”因此，尽管目前Yahoo!在Hadoop中的贡献还是最大，但其占比一直在下降，这在韩轶平看来是件好事，因为现在有更多的公司在为Hadoop做贡献，Hadoop生态系统也日益完善。

目前看来，对Yahoo!老大地位形成挑战的几家势头较猛的公司是Cloudera、Facebook、Hortonworks，还包括国内的企业，比如华为的增长就很快。尽管华为此前从不开源，但现在还能够往Hadoop社区里面做贡献，也着实让人感到惊讶。韩轶平点评道：“这也从侧面反映了开源在中国慢慢得到认可了。”查礼则认为：“我觉得还是Hadoop给它带来了好处，或者至少没有什么坏处。”

## MapR在特定应用场景下做出了贡献

2011年5月，MapR公司推出分布式文件系统和MapReduce引擎——MapR Distribution for Apache Hadoop，号称比开源版本的Hadoop性能高数倍。

对此，韩轶平解释道：“在今年的Hadoop Summit上，我了解到MapR确实在某些特定的应用场景和特定的问题规模下，起到了一定的效果。这样的优化对业界和整个社区绝对是件好事，因为总体上能够推进技术的发展。但我不认为它能够完全解决问题，因为要把Hadoop做成一个通用系统，把所有问题都解决，还有很漫长的路要走。我很欣赏针对一类具体的问题，去提供一类的优化解决方案，这是绝对可能的，因为当整个产业扩大了以后，必然会出现某种程度上的分化或者定制化。但我不觉得它已经本质性地改变了

Hadoop。而对于它公布出来的成绩，我也不怀疑，因为它是针对某些限定条件而得出的。”

查礼也表示认同：“既然MapR要兼容Hadoop，那么它的模式肯定没有创造性或者颠覆性的变化。Hadoop本身有很多能够优化的地方，在某个Benchmark上完全可以做到比Hadoop快很多。做系统其实就是在平衡，舍弃一些特性，就像CAP理论一样，总是有舍有得，不可能设计出一个完美的系统。”

## 2012年Hadoop畅想

2011年，可以看作是Hadoop蓄势待发的准备年，那么2012年，Hadoop又会怎样呢？

韩轶平讲出了他的Hadoop愿景：“第一，完善Hadoop生态系统，而不仅仅局限于Hadoop本身的持续开发、新特征加入和改进。第二，希望进一步根据不同的应用场景、专业领域和应用逻辑，出现更多的中间件和解决方案，因为它们能够帮助Hadoop走出互联网公司，走进更多的公司和行业，从而真正推动整个信息产业和行业的发展。”

查礼也畅想道：“2012年应该是Hadoop向传统信息领域进军的一年。在这个领域，海量数据的问题早就存在，只是没有办法解决或者解决的成本太高，再加之那种‘反正这样的问题同行们也都解决不了’的心态，所以就先搁置。互联网企业已经为Hadoop的应用做出了表率，明年完全有理由期待它在传统信息领域得到更多的应用。”

众所周知，传统行业手中握有非常多的数据，但苦于没有好的方法充分利用起来。通过像Hadoop这样的大数据处理技术，完全可以帮助传统行业创造出更新的价值：一方面可以使其传统业务逻辑得到更有效的实现，另一方面也能够培养出一一种思维和方法，使其慢慢从中发现更多的价值，并进而延伸出更多的价值。此外，Hadoop还可能催生很多其他行业，例如可以将高铁网络上的所有运行数据、轨道数据采集处理以后，进行优化，就会对交通运输的价格体系和物流体系产生深远的影响。

社会总是要前进，需要有更新的产业出现，Hadoop是一个值得信赖的推动引擎。P



# Beyond MapReduce

## 谈2011年风靡的数据流计算系统

文 / 杨栋

在MapReduce计算模型风靡全球之后，无论在工业界还是学术界，数据流计算都将会是下一个热点。本文从深层次对各种典型的数据流计算系统架构及其基于的设计理念进行了剖析。

### 背景与动机

#### 背景

随着社会数据量的日益膨胀，普通服务器组成的计算集群用于处理各种数据应用。在工业领域，像网页检索、机器翻译、广告投放等；在科研领域，像生物复杂分析、自然语言处理、气候模拟预测等。

MapReduce是一种很好的集群并行编程模型，加之拥有强大的开源实现——Hadoop，能满足大部分应用的需求。不过MapReduce只是分布式/并行计算的一个方面，对于交互式计算或者数据流计算，其效果不是很好。数据流计算模型有以下几方面特点。

- 流中的数据元素在线到达，需要实时处理。
- 系统无法控制将要处理的新到达数据元素的顺序，无论这些数据元素是在一个数据流中还是跨多个数据流；也即重放的数据流可能和上次数据流的元素顺序不一致。
- 数据流的潜在大小也许是无穷无尽的。
- 一旦数据流中的某个元素经过处理，要么被丢弃，要么被归档存储。因此，除非该数据被直接存储在内存中，否则将不容易被检索。
- 数据流系统涉及的操作分为有状态和无状态两种，无状态算子包括union、filter等，有状态算子包括bsort、join、aggregate等。有状态算子若执行失败后，其保持的状态会丢失，重放数据流产生的状态和输出不一定和失效前一致，而无状态算子失败后，重放数据流能够构建与之前一致的输出。

数据流计算可以描述为一个个算子（节点）和一条条数据流（边）组成的数据流图。

#### 需求

典型的商用搜索引擎，通常在用户查询响应中提供结构化的Web结果，同时也插入基于流量的点击付费模式的文本广告。为了及时处理用户反馈，需要低延迟、可扩展、高可靠的处理引擎。对于社交网站来说，实时统计和分析用户行为数据，精准地推荐朋友或者及时地反馈圈子动态，都会极大地提升用户体验，增加用户黏性。另外，对于一些反作弊业务来说，尤其是与计费相关的业务，对时效性、可靠性和准确度的要求都会非常高。

#### 挑战

为了保证实时性，许多实时数据流处理系统都是专用系统，不得不面对可靠性、扩展性和伸缩性方面的问题；使用MapReduce的好处在于Hadoop帮助业务屏蔽了底层处理，上层作业不用关心容错和扩容方面的问题，应用升级也很方便；不过基于MapReduce的业务不得不面对处理延迟的问题。

有一种想法是将基于MapReduce的批量处理转为小批量处理，将Hadoop改造成支持常驻内存的工作方式，并改造map和reduce，使其能够在内存中直接传输数据。不过MapReduce框架为批处理做了高度优化，系统典型地通过调度批量任务来操作静态数据，任务不是常驻服务，数据也不是实时流入；而数据流计算的典型范式之一是不确定数据速率的事件流流入系统，系统处理能力必须与事件流量匹配。数据流实时处理的模式决定了要和批处理使用非常不同的架构，试图搭建一个既适合流式计算又适合批处理的通用平台，结果可能会是一个高度复杂的系统，并且最终系统可能对两种计算都不理想。因此，当前业界诞生了许多数据流实时计算

系统来满足各自需求，当然除了延迟，它们需要解决可靠性、扩展性和伸缩性等方面的挑战。

相关工作

Facebook Puma

Facebook实时数据流的用例主要来自站点和网页分析。Facebook Insights允许站长、广告商、应用开发者和网页主根据时间来查看展现/点击/动作计数。计数通过人口统计学逻辑如性别和年龄来区分，还可以查看独立访次数和热点如最受欢迎的网址。

构建Facebook Insights产品后端服务的主要挑战有两方面：一是数据流量非常大，包括Facebook和非Facebook的网站流量；二是用户希望尽快看到摘要，以便立刻知道最受欢迎的文章或最新的游戏。

早期的Facebook通过已有的数据仓库解决方案来处理Insights流量。日志数据流从HTTP服务器产生，通过日志收集传输框架Scribe耗费秒级时间传送到共享存储NFS文件系统，然后通过小时级的Copier/Loader将数据文件上传到Hadoop。数据摘要通过每天例行的流水作业产生，结果定期会更新到前端的Mysql服务器，以便通过OLTP工具产生报表等。Hadoop集群节点有3000个，扩展性和容错性方面的问题能够很好地解决。Copier/Loader其实就是MapReduce作业，能够屏蔽机器故障等问题。流水作业是由基于Hive的类SQL语言开发。如图1所示。

早期的这套系统扩展性很好，能够达到数据中心的部署上限，不过整体的处理延迟较大，从日志产生起1~2天后才能得到最终的报表。如图2所示。

Facebook早期的数据流处理系统Puma如图2所示，PTail将数据以流的方式传递给Puma2，Puma2每秒需要处理百万级的消息，处理多为Aggregation方式的操作，遵循时间序列，涉及的复杂Aggregation操作诸如独立访次、最频繁事件等。对于每条消息，Puma2发送“increment”操作到HBase。考虑到自动负载均衡、自动容错和写入吞吐等因素，Puma选择HBase而不是Mysql作为其存储引擎。Puma2的服务器都是对等的，也即同时可能有多多个Puma2服务器向HBase中修改同一行数据。因此，Facebook为HBase增加了新的功能，支持一条

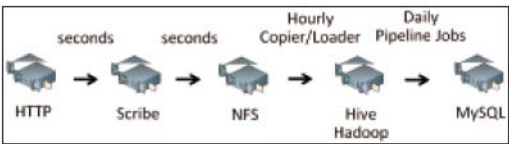


图1 Facebook传统日志处理流程

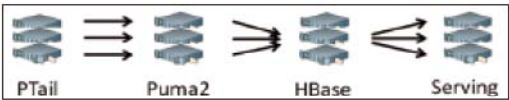


图2 Puma2系统数据处理通路

increment操作修改同行数据的多个列。

Puma2架构非常简单且易于维护，其涉及的状态仅仅是PTail的checkpoint，即上游数据位置，周期性地存储在HBase中。由于是对称结构，集群扩容和机器故障时的处理都很方便。不过，Puma2缺点也很突出：首先，HBase的increment操作非常昂贵，因为它涉及到读和写，而HBase的随机读效率是很差的；其次，复杂Aggregation操作也不好支持，需要在HBase上层维护很多用户代码；再次，Puma2在故障时会产生少量重复数据，因为HBase的increment和PTail的checkpoint并不是一个原子操作。

由于Puma2的缺点限制了整个数据通路的延迟优化，因此Facebook开发了新的Puma3系统。两代系统最大的区别在于Puma3的Aggregation是在本地内存执行的，而不是基于HBase，这使得Puma3的处理延迟很低。为了支持内存中的Aggregation，数据在Puma3中通过Aggregation Key进行分片，分桶策略在传输通道的Calligraphus阶段实施的。Puma3的每个分片都是内存中的哈希表，每个表项对应一个Key及用户定义的Aggregation方法，包括count、sum、avg等等。HBase只作为持久化存储，定期Puma3将内存数据Checkpoint到HBase中，只有当Puma3故障时才从HBase中读相关数据进行重放。由于正常处理逻辑绕开了HBase的读流程，因此整个处理系统的效率大大增加。如图3所示。

Twitter Storm

Twitter首席工程师、Storm作者Nathan Marz最近有一篇很火的文章叫做How to beat the CAP theorem，讲述了他对于数据系统设计的全新理念。为了处理最近几个小时的数据，Twitter需要一个实时系统和批处理系统同时运行。这个实时系统在最近几小时数据上预计算查询函数。要计算一个查询函数，需要查询批处理视图和实时视图，并把它



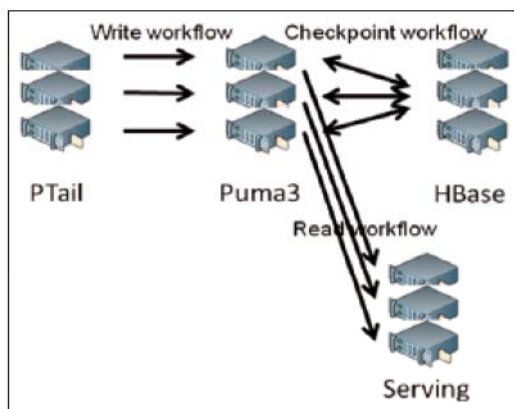


图3 Puma3系统数据处理通路

们合并起来以得到最终数据。进行实时计算的系统就是Storm，它在数据流上进行持续计算，并且对这种流式数据处理提供了有力保障。在批处理层仅需要考虑数据和数据上的查询函数，批处理层因此很好掌控。在实时层，需要使用增量算法和复杂的NoSQL数据库。把所有复杂问题独立到实时层中，对系统的鲁棒性、可靠性有重大贡献。如图4所示。

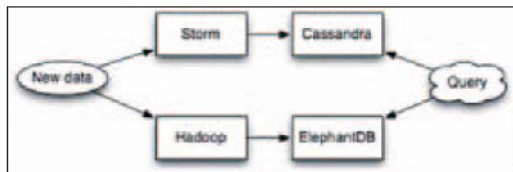


图4 Twitter数据系统分层处理架构

Twitter拥有如此分层的数据处理架构，Hadoop和ElephantDB组成批处理系统，Storm和Cassandra组成实时系统，实时系统处理的结果最终会由批处理系统来修正，正是这个观点使得Storm的设计与众不同。

Storm借鉴了Hadoop的计算模型，Hadoop运行的是一个Job，而Storm运行的是一个Topology。Job是有生命周期的，而Topology是个Service，是个不会停止的Job。Storm的应用场景主要有以下三类。

- 信息流处理 (Stream processing)：Storm可用来实时处理新数据和更新数据库，兼顾容错性和可扩展性。
- 持续计算 (Continuous computation)：Storm可进行持续Query并把结果即时反馈给客户端。比如把Twitter上的热门话题发送到浏览器中。
- 分布式远程序调用 (Distributed RPC)：Storm可用来并行处理密集查询。Storm的拓扑结构是一个等待调用信息的分布函数，当它收到一条调用

信息后，会对查询进行计算，并返回查询结果。例如Distributed RPC可以做并行搜索或者处理大集合的数据。

Storm主要有以下特点。

- 简单的编程模型：类似于MapReduce降低了并行批处理复杂性，Storm降低了实时处理复杂性。
- 支持各种编程语言：可以在Storm之上使用各种编程语言。默认支持Clojure、Java、Ruby和Python。要增加对其他语言的支持，只需实现一个简单的Storm通信协议即可。
- 支持容错：Storm会管理工作进程和节点的故障。
- 支持水平扩展：计算是在多个线程、进程和服务端之间并行进行的。
- 可靠的消息处理：Storm保证每个消息至少得到一次完整处理，任务失败时会负责从消息源重试消息。Storm使用一种比较巧妙的方式判断tuple是否丢失，即发送方和应答方都需要向系统内部的第三方acker异或一个随机数，如果在超时时间内acker上记录的消息随机数为0则认为消息已被正确处理，否则acker将通知spout进行重放。如图5所示。
- 高效消息处理：系统的设计保证了消息能得到快速的处理，使用OMQ作为底层消息队列。

Storm并不提供消息的去重和算子状态的持久化，如果有这方面的需求，需要在应用层面进行处理。这两点也符合Storm的设计初衷，消息重复是其为了容错和保证消息不丢而带来的副作用，但对大多数实时计算来说，并不会要求达到批处理的那种准确度，所以大多数应用都不会对消息重复比较敏感；如果应用对消息重复很敏感，可以在应用层进行去重或者将消息处理定义为幂等操作。

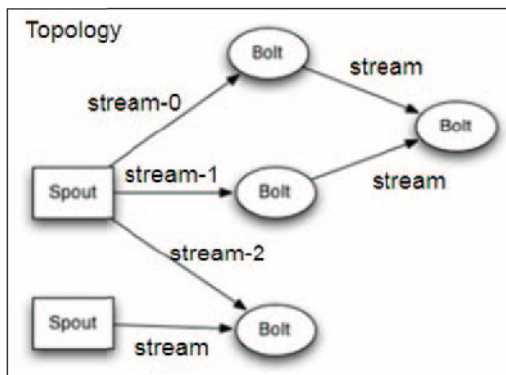


图5 Storm数据流处理示意图

作，再或者由批处理层来修正。

Storm系统并提供任务状态的持久化和高可靠，比如任务failover后的内存状态恢复。这些状态的保存和恢复需要任务自身依赖外围系统来保证，比如依赖一个高可靠的key-value存储系统，或者分布式文件系统等等。正因为不考虑持久化方案，Storm的设计和实现得到大大简化，回头看看它面向的应用，大多数并不涉及状态持久化，有需要的应用层面也可以解决，批处理层也会进行最终的Merge校验，这个设计思路也符合Nathan提出的数据系统分层架构。

Yahoo! S4

Yahoo! S4是一个通用、分布式、可扩展、分区容错、可插拔的流式系统。S4的诞生是为了提高广告点击流量的处理效率，基于S4框架，开发者可以容易地开发面向持续流数据处理的应用。S4的最新版本是alpha version v0.3.0，最近刚成为Apache孵化项目，其设计特点有以下几个方面。

- Actor计算模型：为了能在普通机型构成的集群上进行分布式处理，并且集群内部不使用共享内存，S4架构采用了Actor模式，这种模式提供了封装和地址透明语义，因此在允许应用大规模并发的同时，也提供了简单的编程接口。
- 对等集群架构：除了遵循Actor模式，S4也参照了MapReduce模式。为了简化部署和运维，从而达到更好地稳定性和扩展性，S4采用了对等架构，集群中的所有处理节点都是等同的，没有中心控制。这种架构将使得集群的扩展性很好，处理节点的总数理论上无上限。
- 可插拔体系架构：S4使用Java开发，采用了极富层次的模块化编程，每个通用功能点都尽量抽象出来作为通用模块，且尽可能让各模块实现可定制化。
- 支持部分容错：基于Zookeeper服务的集群管理层将会自动路由事件从失效节点到其他节点。除非显式保存到持久性存储，否则节点故障时，节点上处理事件的状态会丢失。
- 支持面向对象：节点间通信采用“plain old Java objects” (POJOs) 模式，应用开发者不需要写schemas或用哈希表来在节点间发送tuples。

S4的重要应用场景就是点击通过率（CTR）预估这

类应用。据S4的开发者称，在线流量上的实验显示基于S4系统的新CTR计算框架可以在不影响收入的前提下将CTR值提高3%，这主要是通过快速检测低质量的广告并把它们过滤出去而获得的收益。

S4系统提供的低延迟处理能够使得商务广告部门获益，但是潜在的风险也不能忽视，那就是事件流的速率快到一定程度后，S4可能无法处理，会导致事件的丢失。图6显示了8台服务器的测试集群在流量压力测试下的事件丢失情况。

Events per second	Relative Error in CTR	Data Rate
2000	0.0%	2.6 Mbps
3644	0.0%	4.9 Mbps
7268	0.2%	9.7 Mbps
10480	0.4%	14.0 Mbps
12432	0.7%	16.6 Mbps
14900	1.5%	19.9 Mbps
16000	1.7%	21.4 Mbps
20000	4.2%	26.7 Mbps

图6 S4在流量压力测试下的事件丢失情况

模型和挑战

Puma、Storm和S4都是当前比较典型的数据流计算系统，之前讨论了它们各自的设计、实现和应用，我们再通过各个维度的建模视角来分析这些系统，领悟数据流系统的设计原则。本文从处理模型、调度模型等几个方面来度量这几个系统。

分析系统的整体架构

在分析数据流系统模型之前，先来看整个分析系统的整体架构，图7给出了实时计算系统在数据分析大系统中所处的位置。实时计算系统和批处理计算系统同属于计算这个大范畴，批处理计算可以是MapReduce、MPI等，实时计算可以是S4、Storm等，批处理和实时都可以或不依赖统一的资源调度系统。另外，计算系统的I/O，包括中间过程的I/O，都与存储系统HDFS或者传输通道Scribe等交互。计算系统的上层是数据仓库或者计算系统直接和用户交互，交互方式可以是SQL-like或者MR-like等。计算系统的输出结果可以导入数据库，由OLAP工具产生报表等，也可以导向前端OLTP系统，及时反馈用户。

处理模型

在传统的实时数据流处理系统中，业内早期经常使用Queue+Worker的处理方式。系统维护者静态



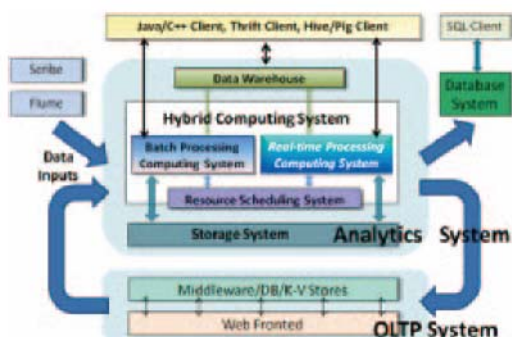


图7 数据分析系统整体架构

配置Worker和Queue的对应关系，即哪些Worker从哪些Queue中读数据，往哪些Queue中写，如果流量或业务增加，需要扩容Queue或Worker时，可能需要重新规划两者的对应关系。为了保证可靠性，Queue通常具有高可用的特性，Worker发来的消息都会被Queue持久化保存，而每个Queue对每条消息都持久化的开销就会有些高，并且会造成消息处理的延迟增大。阿里、百度和腾讯都有基于这种框架的业务处理系统，甚至Facebook早期也是这么处理的，只不过包括Facebook Puma2、支付宝的海狗等系统使用HBase作为Queue，屏蔽了高可用和扩容方面的问题，但低延迟的问题依然会呈现。

为了使得数据流处理系统的延迟变低，所有数据计算必须在内存中执行，并且流动数据不能每条都通过Queue进行持久化，这样子数据处理的高可用又变成了亟需解决的问题。大部分数据流处理的高可用技术基于故障恢复，如果服务器故障，一组预先定义的备份机器将接管失败的执行。通常基于故障恢复的高可用方法有三类（图8对这三种高可用方法进行了比较）。

■ **Passive Standby:** 在该方式下，每个primary定期做checkpoints（即拷贝从上个checkpoint到当前状态的改变）到它的backup。Passive Standby方式广泛试用于局域网集群，它能够经受高负载情况，却降低了恢复速度。一些已有技术将每个服务器上的计算进行拆分，将拆分后的部分备份到多个服务器。基于这种分布式备份，能够实现恢复的并行化，大大减少恢复时间。

■ **Active Standby:** 在该方式下，每个backup也会接收并处理输出数据，和primary并行，在广域网应用环境下，是个好的选择。比较Passive Standby，

Active Standby方式保证了更快的恢复速度，低负载情况下所有backup使用和primary相同的资源。通过使用分配给tuples的序列数，Flux技术实现了无损和副本自由的故障恢复语义。Flux允许副本之间松耦合，一个副本能够比其他执行得更快，导致在很多时候执行交迭。Balazinska等人研究Active Standby方法的变体，能够处理网络故障和分割。

■ **Upstream Backup:** 在该方式下，每个primary记录它的输出数据到日志，这样即使下游primary发生故障，backup也能通过重放日志数据来从头重新构建丢失的状态。Upstream Backup在无故障期间执行效率很高，因为其backup一致保持空闲状态。然而，它可能会花费较长时间来恢复状态较大的计算。例如，恢复一个带有30分钟窗口的Aggregation算子，需要重新处理30分钟内的所有tuples。对于那些算子的状态很小，并且资源紧缺，故障发生稀少的情况，Upstream Backup是个好的选择。

## 调度模型

在调度模型方面，Storm拥有自己的主节点Nimbus，而Puma和S4号称对称架构，没有中心节点，但我有理由详细，这些系统未来也会有自己的主节点。对称结构没有单点，扩展性理论上无限，容错和负载均衡需要依赖分布式协议，例如借助Zookeeper；而主从结构在实现故障恢复和负载均衡上更加容易，缺点是存在单点，可能会造成性能或稳定性方面的问题。仔细分析下，其实数据流系统的主节点不存在单点问题：

■ **主节点无状态**，可以有多个Standby节点，都向Zookeeper注册，当主节点故障后，自动切换到Standby节点，这点和BigTable类的系统是类似的。

■ **主节点并无性能瓶颈**，像MapReduce这类批处理系统，由于任务是有生命周期的，因此主节点需要频繁调度任务，规模增大时容易造成主节点的调度压力增大；而数据流系统的任务常驻内存，一旦执行就不退出，即只有启动或故障时才会调度，因此调度并无压力。

Storm将每个任务都注册到Zookeeper，由Zookeeper去检测任务的存活，进而通知主节点；也有一些系统通过任务的本地守护进程感知故障并向主节点汇报，两者并无本质差异。

	Passive Standby	Active Standby	Upstream Backup
Processing Speed	Low	High	High
Recovery Speed	Med	High	Low
Resource Usage	Med	High	Low
Precise Recovery	Yes	No	No
Extension	LAN	WAN	LAN/WAN

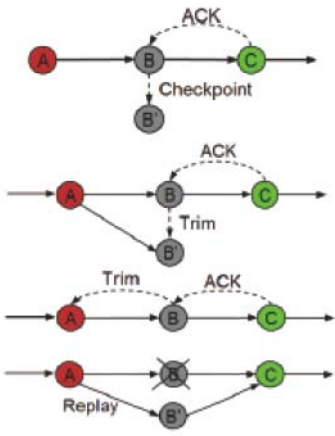


图8 数据流计算系统的三种高可用技术

Feature	Puma	Storm	S4	Comments
Development	Java	Clojure	Java	Different systems towards different application requirement
HA	Primary Standby	Upstream Backup	Primary Standby	
Precise Recovery	No	No	No	
Architecture	Symmetric	Master-Slave	Symmetric	
Resource Utilization	Low	High	Low	
Recovery Time	Short	Long	Long	
State Persistence	Yes	No	Yes	
De-dup	Yes	No	No	
★ Universal Data Stream Processing System				

图9 Puma、Storm和S4三种数据流计算系统对比

数据流系统有主节点之后，任务调度和故障处理就变得方便了许多，将哪些任务调度到哪些机器去执行，这依赖于系统状态和机器的资源使用等因素。

另外，数据流系统需要考虑负载分流方面的问题，即当流量增加时，如果将负载均匀地分流到集群的处理节点。一般有几种做法，根据流量可以动态地对任务进行分裂，分裂后的多个任务再进行调度；也可以先静态地配置任务的粒度，让每个相关任务处理的流量较少，当发现流量增加时进行任务迁移。后者实现简单，但对资源的使用存在一些浪费；前者实现复杂，对无状态的任务拆分可行，但如果用户定义的任务有自身状态，那么拆分用户状态是非常困难的事情。

结束语

实时数据流计算是近年来分布式/并行计算领域研究和实践的重点，诞生了多个具有代表性的数据

流计算系统，用于解决实际生产问题和进行学术研究。不同的系统满足不同应用的需求，系统并无好坏之分，关键在于服务的对象。对于各个公司的业务需求来说，那个系统就是它们最需要的系统。图9对Puma、Storm和S4这三个分别适用于站点统计、社交网络和广告投放等应用的数据流计算系统，从开发语言、高可用机制、支持精确恢复、主从架构、资源利用率、恢复时间、支持状态持久化及支持去重等几个方面对这三种系统进行了对比。

可以看出，为了高效开发，两个系统使用Java，另一种系统使用函数式编程语言Clojure；高可用方案，有两个系统使用Passive Standby方式，系统恢复时间可控，但系统复杂度增加，资源使用率也较低，因为需要一些机器用来当备机；而Storm选择了更简单可行的上游回放方式，资源使用率高，就是恢复时间可能稍长些；Puma和S4都支持状态持久化，但S4目前不支持数据去重，未来可能会实现；三个系统都做不到精确恢复，即恢复后的执行结果和无故障发生时保持一致，因为即使是Passive Standby方式，也只是定期Checkpoint，并没有跟踪每条消息的执行。商用的StreamBase支持精确恢复，这主要应用于金融领域。

实时数据流计算在科研领域已有多年的研究，诞生了像Borealis、DSMS等系统。由于网络数据的不断膨胀和用户需求的不断涌现，近年来互联网企业开始广泛研究和使用的数据流处理，诞生了Puma、Storm和S4等系统。正是需求推动了技术的进步，技术革新也影响着用户需求，数据流计算领域的发展才刚刚起步，未来一定更加辉煌。



杨栋  
百度分布式高级研发工程师，从事Hypertable、Hadoop及流式计算的研究和开发。

责任编辑：董世晓 (dongsx@csdn.net)



# 谈保险公司数据服务平台的建立

文 / 张丽芳

本文从作者所在保险公司建立数据服务平台的实践出发，阐述了该公司建立数据服务平台的背景、选取的各种工具和工具的使用效果，以及建立数据服务平台的具体步骤。

近年来，我一直从事保险公司有关IT项目的开发工作。前年，我参与了公司数据服务平台的建立。在项目中，我担任系统分析设计以及数据库管理等工作。

这家保险公司是一家历史悠久的企业，分为总部、省级和地市分公司三级。

早期该公司的生产系统为地市级逻辑集中，各地市IT人员根据本地市需求，针对生产系统提取数据制作报表供本地市领导决策，省公司没有生产系统。

随后几年，随着该保险公司省级集中步伐的加快以及新的生产系统的上线，地市级公司不再拥有生产系统的数据，所有生产系统全部变为省级集中。

最近两年，全国金融系统大集中的趋势越来越强烈，因此，我们目前主要的核心系统已经均为全国物理集中。

这样就形成了新的问题：省级分公司再没有权限在本省分公司的生产库上提取数据，因而无法得知本省的业务发展情况；对省公司来说，省公司领导需要将各种生产系统的数据进行汇总合并后得到总的业务数据，从而进行决策分析；另外，省公司的其他部门如个险部、团险部、银保部、业务管理部、企划部等也需要掌握自己感兴趣的业务数据，从而指导本部门业务发展方向。

公司内数据的真正状况是分散而非集成的，数据不一致问题、外部数据和非结构化数据问题都难以解决。针对这样的情况，我们项目小组建立了数据服务平台。

## 需求调研

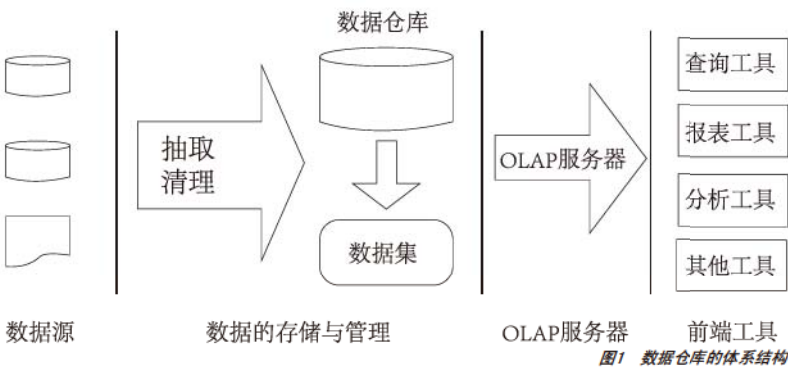
数据服务平台的核心是数据仓库。数据仓库将不同介质、不同组织方式的数据集成转换而成为一个一致的分析型数据环境，为不同来源的数据提供一致的数据视图。它的最大优点在于能把企业网络中不同信息岛上的数据集中到一起，存储在一个单一的、集成的数据库中，并提供各种手段对数据进行统计、分析，且允许企业的各个部门共享数据，为企业更快、更好地做出决策提供更加准确、完整的信息。

首先，需要了解地市分公司、省公司领导以及省公司其他部门对数据的需求。在调研过程中，我们项目小组将需求方召集在一起，召开了多次需求研讨会，也多次下到地市，调研他们的真实需求。地市分公司列出了自己感兴趣的报表数据。对领导层，我们详细询问了领导所关心的业务数据和指标。同时省公司其他部门也提出了自己感兴趣的数据需求。

其次，根据这些需求，我们多次和用户进行沟通，明确每个需求的具体含义，以及基于这些需求可能产生的新需求。好的一点是，项目方人员都是公司内部员工，长期的工作经验使部门间的沟通过程比较顺畅，理解方面也不存在大的偏差。需求调研过程比较顺利。

## 数据仓库技术选型

图1是数据仓库的体系结构。下面将重点讨论数据仓库的建立和前端工具的使用。



数据库采用的是SQL Server。在整个环境中，集成了如下服务：数据库服务、ETL工具（Integration Services）、报表工具（Reporting Services）、分析工具（Analysis Services），其中许多BI工具都是很成熟的。应用前端基于.NET平台，报表制作使用Reporting Services。在微软平台上，采用.NET技术一定是优于Java的。

在选择建设数据仓库具体的技术方案时，我们进行了认真的思考和研究。

### 数据库

数据库的选型上，是像我们业务系统一样采用informix数据库，还是采用基于Windows的SQL Server等数据库？

一方面，采用informix作为数据库有很多优点：长久以来，我们公司的生产系统都是用基于UNIX的informix作为数据库，长期的应用和维护使我们的IT人员对informix都很熟悉，积累了一大批技术人才。

但另一方面，informix也有很多不尽如人意的地方：首先是性能方面，很多核心参数都需要手工调整，而面对建设数据仓库这样一个数据量大并且我们从未涉及的项目（以前都是涉及事务级数据库，从未涉及过数据仓库），大家对它的性能调优等还是存在一定的担心。

另外，informix的发展也有点跟不上现代数据库的发展，比如它的单表最大只支持2GB，如果表超大就必须进行转储或分片；而且其表的extend需要经常关注，对于entend过大的表则需要手工修改extend，下载表数据，再重新上载表数据。

Informix的这些特点，使我们认为它不适合建设数据仓库。

而微软基于Windows的SQL Server数据库则无需像informix一样关注那些系统级的设置。数据库采用SQL Server，加上我们公司诸多的Windows服务器以及与微软的良好合作关系，将数据仓库建设在SQL Server上应该会比较合理的选择。由于选择了SQL Server作为数据库，数据抽取和清理的工具，我们选择了SQL Server自带的ETL工具Integration Services（SSIS）。

### 数据抽取

在数据抽取方面，我们将该数据仓库分为四个数据层。

■ **镜像库。**在各个生产系统上建立了与生产库同名的镜像库，镜像库中相应的表建立trigger，记录生产系统表的变化（增、删、改），将这些变化写入到与生产库同构的镜像库相应的同名表中。镜像库与生产系统都在informix端，建立在不同的数据库online上。针对每个生产库都对一个镜像库，镜像库与生产系统的对应表是同构关系。

■ **增量层。**增量层部署在SQL Server端。通过SSIS工具，将镜像库的数据进行抽取转换和装载，写入各个生产系统相应的增量层中。增量数据库与生产系统也是对应关系，有几类业务系统就会有几个增量数据库。增量数据库与业务系统是同构关系，是非集成的。由于有了增量数据库层，所以数据转换不再与业务系统相关，整个过程会比较快捷。

■ **基础层（Base层）。**在增量层之上，建立一个Base层，包括合同类、收付类、客户类、理赔类等几类表。基础层只有一个库，它对增量层的不同业务数据进行了融合和汇总，存放融合的基于操作级的数据。

■ **汇总层（Cube层）。**汇总层是在基础层或增量层的基础上构建的，主要是针对前台的报表应用建立统计模型。它对数据进行了抽象和加工，并不是基于操作型数据的。它适合于报表生成和数据分析。



通过DTS抽取工具，我们将生产库的数据进行抽取和清理。抽取分为存量抽取和增量抽取。系统初始化时，进行一次存量抽取。在镜像库和增量库表中设计有时间戳，记录最后一次更新时间和id，这样就实现了增量抽取机制。

经过初始化一个多星期的抽取，生产库存量数据已经全部抽取成功。后期在日常的增量抽取过程中，出现了一些问题。比如生产库有很多垃圾数据，在抽取时会报错。

为了解决这个问题，我们在DTS设计了很多容错机制。比如SQL Server在处理日期时，对于小于1753-01-01的，会报错退出。而生产库偶尔就会碰到操作员手误录入日期错误的。我们就在抽取DTS中加入了日期判断，小于1753-01-01的就统一置为1900-01-01。这是一种简单的容错机制，对于要求不是很高的应用基本可以满足。对于另外一些比较重要的日期，在生产库端就错误的，还是需要联系生产库的管理员，从生产库源头进行修改，修改之后再行抽取。

## 前端工具

在前端工具的选择上，最初想到的是ASP。因为过去的几年中我们的IT人员平时使用ASP很多，做过各种大小程序的开发工作，用起来可以说是得心应手。

但经过和Reporting Services的斟酌对比，我们还是放弃ASP，而使用了微软的Reporting Service。对于我们初次接触BI的人来说，Reporting Service部署更简单方便、上手更容易，实现速度更快。而且Reporting Service是集成在SQL Server环境中的，也更为便捷。

Reporting Service是一个完整的基于服务器的平台，可以建立、管理、发布传统的基于纸张的报表或者交互的、基于Web的报表。在Reporting Services中生成的报表包括交互功能和基于Web的功能，在外观和功能上超越了传统的报表。例如，深化报表允许在数据层间进行导航；参数化报表支持在运行时对内容进行筛选（允许用户选择机构以及各种预定义的条件以实现对报表内容

的筛选从而达到特定用户的特殊需求）。最终的报表是基于Web的，可以实现向下钻取，支持多种报表导出格式，尤其对Excel的导出支持得更好。

## 结束语

整个项目历时近两年，从这两年的情况来看，系统是成功的。在金融系统数据集中的大环境下，数据服务平台能很好地完成各类用户对数据的需求，能对领导层提供感兴趣的数据分析和决策支持，并提供各类基础数据为日后其他增值应用提供服务。但数据仓库的建设不是一劳永逸的，而是一个不断完善和发展的过程。

在我们的部署过程中，数据仓库存储量很大，可以从如下方面考虑效率的提升。

- 使用更多的PC服务器来部署，分担单个服务器所承载的压力。
- 使用高速存储设备。目前我们采用SAN存储，数据经过光纤（而非局域网）传输，可实现数据的高速可靠传输，并能实现灵活的存储设备配置要求。
- 与微软合作，优化SQL Server的性能。请微软讲师专门做了一系列关于SQL Server性能调优和优化的培训。
- 使用64位的PC服务器。在项目之初，由于设备不足，使用的还是32位的PC服务器。中后期已经陆续更换为64位PC服务器，整体性能得到了很大的提升。P



张丽芳

系统分析师，就职于中国人寿保险股份有限公司陕西省分公司信息技术部，从事informix、SQL Server、Oracle方面的开发和研究。

责任编辑：董世晓（dongsx@csdn.net）

# 前端技术这一年

文 / 陈广琛

前端工程师的2011年惊喜不断。很多前端技术如同寒武纪大爆发，得到了长足的发展与丰富，本文从国外、国内两个角度，回顾了前端技术的热点话题。

## 国外前端

对前端工程师来说，2011年是最适合创作的一年。浏览器的快速迭代，把HTML5的最新草案技术推送到用户的桌面，甚至是手机上，前端工程师不用再对着RFC兴叹“这么好的概念我什么时候才能用上啊？”大部分最新的HTML5技术，都能在Google Chrome或Mobile Safari上找到实现。担心今天还是Nightly Build的功能不知道何时才能发布？六周后就能发布，全球Google Chrome及Mozilla Firefox用户都能用上。2011年，Mozilla开始把Firefox的发布周期缩短为六周，Microsoft的Internet Explorer团队因此不再送庆祝蛋糕。跟HTML5相关的另一个重要事件是Adobe宣布停止开发用于移动设备的Flash，这意味着连Adobe也意识到要在移动设备上实现丰富的交互界面，HTML5是更划算的解决方案。

Node.js的流行让前端工程师不再“Think inside the Box”（“Box”指浏览器的框框）。想写个Web Server的原型？不需要懂PHP也不需要Apache，直接用JavaScript写一个；想写JavaScript打包压缩工具？不需要懂Java或Rhino，直接调用Uglify-JS就可以，这是个完全用JavaScript写的JavaScript解析器，源代码完全开放供随时下载；觉得XML配置难以手工维护？那就换成JSON配置吧，一句JSON.parse()就能完成读取。Node.js完全打破了前端工程师原有的浏览器围墙，让前端工程师有机会探索更广阔的天地，也为前端工程师提供了巨大的创作空间。

说到创作空间，我们必须感谢一下Github为协同式创作带来的便捷性。没有Github也能创作，但

你做你的，我做我的，谁也不知道对方做了什么，也就没有了互相借鉴的可能性。Github使得创作由个人行为转变成了社会行为。喜欢一个作品，可以fork一下，然后基于此作品继续创作；不喜欢这个作品，仍然可以fork一下，然后把不喜欢的部分全部改掉。

Github可以说是为Node.js的发展提供了强大的推动力。Node.js跟Ruby和Python类似，拥有自己的包管理系统，叫做NPM（Node Package Manager）。2010年时，NPM上面的包还十分有限。经过了2011年如同寒武纪般的大爆发，现在基本上能想得到的功能，在NPM上面都有对应的包了。考虑到JavaScript开源的天性，这些包基本上都在Github上开源了。很多2011年之前就已经存在的JavaScript开源项目，也把源代码封装为NPM包。这构成了良好的生态圈——想要什么功能的包，都可以直接去社区里找现成的来用，在使用的过程中或许会改良这个包里面的模块，例如修复缺陷、增加功能和性能优化，然后可以把自己的版本再发布到社区。

## 国外项目

2011年里得到迅猛发展的另一个领域是PaaS。Heroku宣布支持Node.js，Cloud Foundry第一时间支持了Node.js，连Windows Azure最近都说支持Node.js了。还记得过去折腾LAMP有多复杂吗？至少要懂好几门不同的编程语言，包括PHP、SQL或者JavaScript，还要会配置Linux、Apache和MySQL，还要租VPS才能把自己的作品展示给别人。从2011年开始，只要会JavaScript就可以发挥自己的想象力，编写头脑里的网站或服务，然后



一句git push即可完成部署，还保证服务稳定性。

## 国内前端

国内方面，2011年前端没有太多突破性进展。考虑到近几年来国内热点都是移动互联网，焦点都转移到“移动”上面来了，对前端的关注还不如对原生移动客户端的多。与此同时，国内大多数的前端工程师还在与Internet Explorer做搏斗，对HTML5的爱只能停留于幻灯片上，难以落实到代码中。

尽管国内前端工程师对HTML5新技术的涌现纷纷表示“影响不大”，Github的开源风倒是吹到国内来了。过去SourceForge和Google Code的Project Hosting满屏幕英语、操作还很复杂，没能引爆国内的开源社区，这次Github通过便捷的操作方式做到了。国内的前端项目纷纷通过Github开源，各大公司的招聘思路也顺应着发生了变化——“你曾经参与过开源项目吗？你曾经向我们公司的开源项目贡献过代码吗？”不懂英文不懂项目管理不再是问题，只要对国内的开源项目感兴趣都可以在Github上fork一下，这些项目大多都有完善的中文文档。

## 国内项目

国内值得关注的项目包括SeaJS、Tangram、QWrap、Kissy和Jscex等。SeaJS为包加载工具，能够在调试时动态加载不同的包，也能在部署时合并多个包。它自带一个包管理工具：佳作SPM（SeaJS Package Manager）。Tangram是百度的前端库，拥有函数粒度的可定制性，同时有众多子项目。QWrap是Tangram里很有特色的子项目，以面向对象的方式进行调用，但函数签名又有点函数式语言的风范。Kissy是淘宝的前端库，拥有丰富的界面组件。Jscex是能让开发者以同步方式编写异步代码的库，使得代码不再出现混乱的回调嵌套。

2011年，Node.js的各式活动在国内也进行得如火如荼，同时阿里云也宣布了对Node.js的支持。经过一年的发展，国内的Node.js社区已有一群活跃



开发者，他们大多数是英语还不错的Github活跃用户，一边通过开源的包学习别人使用Node.js的最佳实践，一边根据自己的需求或者是创意进行再创作。尽管国内还缺乏热门作品，不过开源利己利人的意识已经被前端工程师广泛接受，大家不再觉得自己的代码只有自己在用，转而思考能否把自己的代码整理出来，贡献给更多的社区开发者。

## 结语

相信2011年是让人充满满足感的一年，满足了对新技术的好奇心，满足了学习新技术的热情，满足了使用新技术进行创作的成就感。P



陈广琛

网名Cat Chen，豌豆荚前端架构师，曾任百度前端研发部高级工程师。

责任编辑：高松（gaosong@csdn.net）



主持人: 张银奎

《软件调试》一书作者, 从事软件开发和研究10余年, 对IA-32架构、操作系统内核、虚拟技术, 尤其对软件调试有较深入的研究。翻译(合译)作品包括《数据挖掘原理》、《机器学习》、《人工智能: 复杂问题求解的结构和策略》、《观止——微软创建NT和未来的夺命狂奔》等。

# SDK安装程序卡壳之谜

## 兼谈函数的异常出口(下)

书接上回, 本期继续探索Windows 7 SDK安装程序卡壳之谜, 彻底还原文件下载线程“误入罗网”的来龙去脉。

### 溯本追源

上一期, 我们定位到了文件下载线程中有错误发生。经过分析, 我们怀疑这个线程陷入了死循环, 无法脱身, 导致整个进程停滞不前。是否真的如此呢? 如上期开头所说的, 软件的根本在代码, 从代码层次理解问题往往可以事半功倍。那就看代码吧! 看什么代码呢? 源代码最好理解, 可是没有啊。因为这个SDK安装程序不是我们开发的, 也不是开源的。那是不是就没办法看代码了呢? 这个问题难倒了很多, 让他们不得不改换方向。但事实上有很多方法可以跨过这个障碍。

首先, 退一万步讲, 任何时候总是可以看汇编代码的, 使用调试器或其他工具总是可以得到反汇编出来的汇编语言代码。但有些读者可能会说, 什么年代了, 还看汇编啊, 太恐怖了吧? 的确有些程序员一见到汇编就望而生畏, 其实大可不必。事实上汇编语言并没有想象得那么可怕, 特别是在大家经常工作的x86架构的PC系统上, 因为x86指令具有非常强的表达力(两三个字节的机器码就可以表达一个循环操作), 反汇编出来的程序仍然具有很好的可读性。其次, 对于Java和.NET这些动态语言所开发的程序, 一种非常简单的方法就是使用反编译工具将其逆向成中间语言, 得到的代码大多时候与源代码都有着非常高的相似度, 除非开发者为了反调试而故意做过处理。拿我们眼

下面对的问题来说, 对于使用.NET技术开发的程序, SDK中就包含了一个非常好的反编译工具, 叫ILDASM, 是Intermediate Language Disassembler的缩写。下面我们就用它来反汇编SDK安装程序。

使用文件浏览器, 浏览到SDK的程序目录, 比如, 机器上有多个SDK实例, 其中一个的路径是c:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\Bin\, 找到ILDASM程序后双击启动, 然后通过File>Open菜单调出打开文件对话框, 选择SDK安装程序SDKSetup.exe, 点击确定后, ILDASM便会解析文件内容, 然后以树状图的形式显示出程序中的清单信息以及各个类, 找到感兴趣的类, 点击展开, 便可以看到的类的属性和方法, 比如图1显示的便是我们希望深入了解的DownloadUtils类的详细信息。

回忆上期内容, 我们曾经使用两种方式来观察下载线程的执行过程(栈回溯), 使用sos扩展得到的栈回溯信息中(上期图3), 位于最上方的方法是DownloadUtil类的Dispose方法, 即:

```
SDKSetup.DownloadUtil.Dispose(Boolean)
SDKSetup.DownloadThread.DownloadThreadProc()
System.IO.__Error.WinIOError(Int32,
System.String)
System.IO.FileInfo.get_Length()
SDKSetup.DownloadUtil.DownloadFileAsync
cUsingAsyncFileIO(...)
SDKSetup.DownloadUtil.DownloadFileAsync(...)
SDKSetup.DownloadUtil.DownloadFile(...)
SDKSetup.DownloadThread.DownloadThreadProc()
```

在使用k命令直接产生的本地代码栈



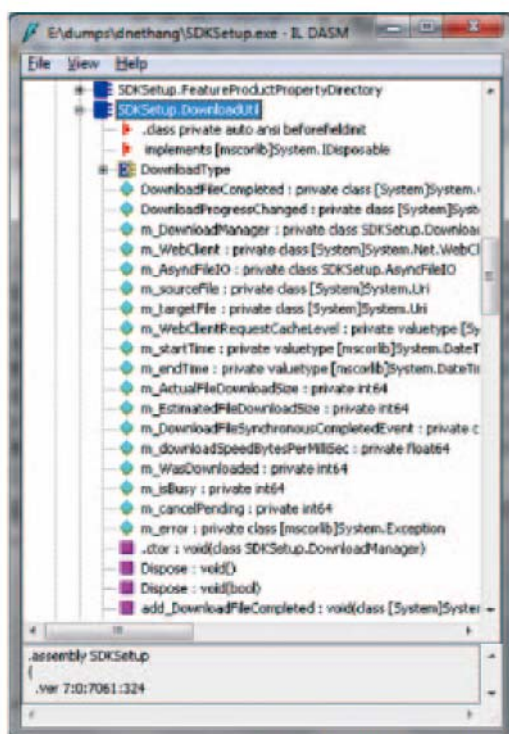


图1 使用ILDASM观察SDK安装程序

回溯信息中（上期图2），位于最上方的是NtDelayExecution方法，即：

```
ntdll!NtDelayExecution+0xa
KERNELBASE!SleepEx+0xab
mscorwks!EESleepEx+0x2d
mscorlib!Thread::UserSleep+0x71
mscorlib!ThreadNative::Sleep+0xf9
0x7ff`001c0dbd
0x7ff`002f4c2d
```

结合这两个信息，我们推测Dispose方法中可能在反复调用Sleep方法，在死循环。为了验证这个推测，现在就用ILDASM来反汇编Dispose方法。从图1可以看到，名字叫Dispose的方法有两个，根据栈回溯，可以判断下载线程调用的是带参数的一个，也就是图1中靠下方的一个，双击这个方法，ILDASM便会反汇编这个方法，并把结果显示在一个弹出窗口中，如图2所示。

## 理解IL

下面我们来理解一下图2中使用中间语言（IL）所表达的Dispose方法。开头和结尾很好理解，我们只看中间的“正文”部分。“正文”部分大体显示为三列，最左边是该行指令相当于这个方法起始地址的字节偏移，表示为IL\_XXXX的形式，比如第一条指令的偏移是0，最后一条指令的偏移是

0x38，整个方法共有0x39（57）个字节。第二列和第三列分别是指令的操作符和操作数。有些指令没有操作数，因此第三列便是空的，比如代表返回到父方法的ret指令。

如何理解每一条指令的含义呢？有三种方法，按照从浅到深的顺序，第一种是根据英文缩写来推测，也即是猜，比如第一条指令ldarg.1中的ldarg是Load Argument的缩写，意思是加载参数，点后面的1代表的是第一个参数，参数是从0号开始算的，但是0号代表的是this，比如IL\_0004出的ldarg.0就是加载this指针，然后调用这个类的方法。因为IL指令是基于栈的，因此ldarg都是把参数加载到栈中。

第二种方法是搜索网络，比如以下wiki页面上有所有IL指令的清单，每条指令有个简短的解释。

[http://en.wikipedia.org/wiki/List\\_of\\_CIL\\_instructions](http://en.wikipedia.org/wiki/List_of_CIL_instructions)

第三种方法是阅读正式文档。因为微软多年前就以工业标准的形式公开了IL的详细定义，因此在名为ECMA-335的标准文档中可以找到所有IL指令的详细定义。通过下面的网页可以下载这个文档，目前是2010年12月发布的第5版，共有565页。

<http://www.ecma-international.org/publications/standards/Ecma-335.htm>

有了上面的基础，便很好理解图2中的代码了。简单来说，第一条指令是把参数disposing加载到

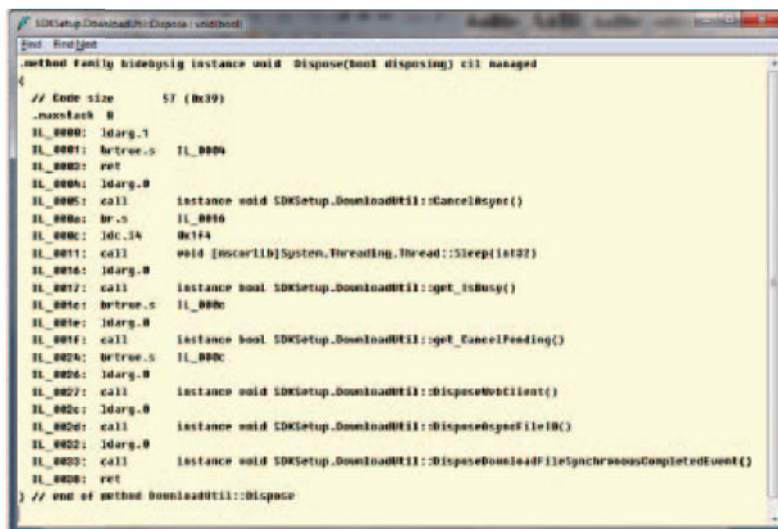


图2 死循环所属方法的中间代码

栈上，接下来brtrue指令是一条条件转移指令，branch on non-false or non-null，也就是判断栈上的操作数disposing的值，如果为真就跳转到IL\_0004，不然顺序执行下面的ret返回了。

IL\_0004处的指令是ldarg.0，也就是把this指针加载到栈上，然后使用call指令调用同一个类的CancelAsync方法，目的是取消可能还没有完成的异步（下载）操作。

接下来的br.s是无条件转移指令，也就是转移到IL\_0016处。

IL\_0016和0017处的两条指令是调用get\_IsBusy方法。它们下面的brtrue.s是判断放在栈上的返回值是否为真，如果为真就向上转移到IL\_000C去调用Sleep。如果不为真则顺序向下执行。

于是，可以把这一段指令翻译为下面的伪代码：

```
if(disposing)
    return;
CancelAsync();
do
{
    Sleep(500);
}while(get_IsBusy());
```

如此看来，这里真的有一个可能无休止调用Sleep函数的死循环，唯一的条件是get\_IsBusy方法一

直返回真。那么get\_IsBusy方法是否会一直返回真呢？不妨看一下图3给出的get\_IsBusy方法的中间代码。

正如名字暗示的那样，get\_IsBusy方法的实现很简单，简单理解就是返回m\_isBusy这个字段的值。其中的ldflda指令代表的是load field address，含义是将对象成员的地址加载到栈上。接下来是以多线程安全的互锁方式读这个字段的值，然后将其从64位整数转换为布尔类型。

这样看来，如果m\_isBusy字段为真，并且没有人清理，那么就可能导致Dispose方法中发生死循环。m\_isBusy字段在“事发”时是否为真呢？使用上期介绍过的ldso和ldo命令看下便知。如图4。

哦，真的为1。看来我们的推测是完全正确的。Dispose方法真的因为m\_isBusy字段为真而发生了死循环。

深挖m\_isBusy字段

如此看来，如果在执行Dispose方法时m\_isBusy字段为真，那么便可能导致死循环，这样说的前提是其他线程不会来修改这个字段的值。这个前提是否成立呢？还是要看代码。这次看的重点是最终m\_isBusy字段的取值。

可能修改m\_isBusy字段的情况有两种，一种是对象本身的方法直接访问和修改m\_isBusy字段，另一种可能是其他对象的方法通过set\_isBusy方法来间接修改这个字段。

在ILDASM中选择Dump功能将安装程序的所有中间代码转存为文件，然后在IL代码文件中搜索m\_isBusy和set\_isBusy来寻找上面两种情况（注意排除其他类包含同名字段的情况）。使用这种方法可以得出m\_isBusy字段的

- 在构造函数中，m\_isBusy被初始化为0。
- 在DownloadUtil::DownloadFileAsync方法的前半部分，m\_isBusy被以互锁交换的方式设置为1。
- DownloadUtil::internal\_DownloadFileCompleted方法的finally块中，有对set\_isBusy方法的调用，这也是整个程序中对set\_isBusy方法的唯一调用。

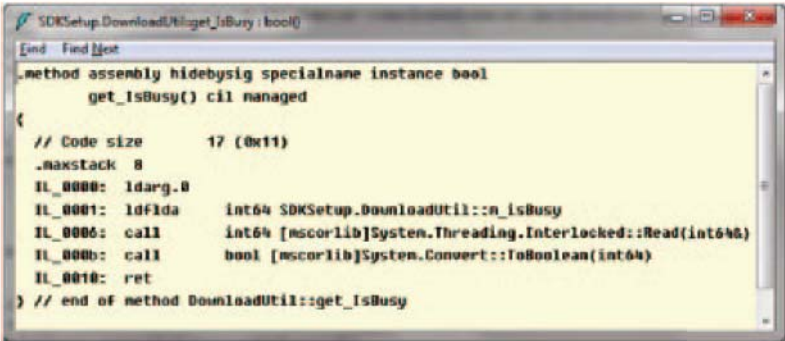


图3 set\_IsBusy方法的中间代码

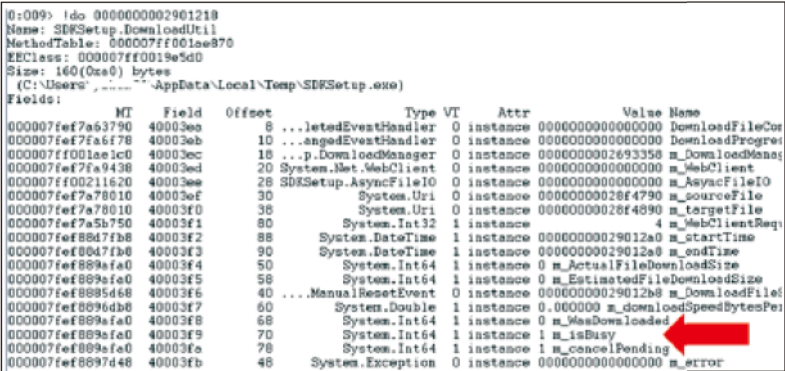


图4 观察m\_isBusy字段的取值



搜索internal\_DownloadFileCompleted, 容易看出, 它会被注册为异步下载操作完成事件的回调函数。

```
ldftn         instance internal_
DownloadFileCompleted()
newobj        instance void
AsyncCompletedEventHandler::ctor()
callvirt      instance void
SDKSetup.AsyncFileIO::add_CopyFileComple
ted(AsyncCompletedEventHandler)
```

这样分析后, 我们便可以推测出m\_isBusy的用途, 正常情况下, 它的变化逻辑是这样的。

DownloadFileAsync方法将其设为1, 代表下载线程进入忙状态。之后, DownloadFileAsync方法调用DownloadFileAsyncUsingAsyncFileIO。后者内部调用CreateAsyncFileIO来创建SDKSetup.AsyncFileIO对象, 目的是以异步方式在新的工作线程中下载文件。创建AsyncFileIO对象时, 会将internal\_DownloadFileCompleted方法注册为回调函数, 以便在下载完成时得到通知。创建对象后, DownloadFileAsyncUsingAsyncFileIO方法调用AsyncFileIO对象的CopyFileAsync方法启动文件下载, 随即便返回。如此看来, 似乎很“完美”。

## 过程“重现”

如果一切正常, 那么以上设计的确可以很好工作。文件下载完成后回调函数中清除Busy标志, 然后执行Dispose方法就不会死循环了。但是有时还是会有意外情况发生。比如我们现在讨论的情况。长话短说, 到了彻底揭晓答案的时候。故事是这样发生的。

首先, 在DownloadFileAsyncUsingAsyncFileIO方法调用get\_Length前一切还都是正常的, 从栈回溯来看, 以下调用序列都是正常的。

```
System.IO.FileInfo.get_Length()
SDKSetup.DownloadUtil.DownloadFileAsyncUs
ingAsyncFileIO(...)
SDKSetup.DownloadUtil.DownloadFileAsync(...)
SDKSetup.DownloadUtil.DownloadFile(...)
SDKSetup.DownloadThread.DownloadThreadProc()
```

但是已经埋下了一个隐患, 那就是在DownloadFileAsync方法中已经将m\_isBusy值1了。而且问题就从get\_Length方法开始了, 因为指定文件不存在, 所以抛出了一个I/O异常, 因为DownloadFileAsyncUsingAsyncFileIO方法中没有编写异常捕捉代码(catch), 所以错过了处理这个异常的最佳时机。异常一旦抛出, 就要有人处理。在当前的方法中找不到, 系统(异常分发函数)便继续在当前方法的父方法中寻找, 但是也没有, 于是继续在父方法的父方法中找, 也没有。当寻找继续到DownloadThreadProc方法时, 终于找到了, 是一个try{ }finally结构。这意味着CPU需要从get\_Length方法“飞回”到DownloadThreadProc方法。说飞回, 是因为不再像正常函数返回那样, 一级级返回, 而是直接飞跃。可巧, 在finally结构中“不幸”包含了一个对DownloadUtil.Dispose方法的调用, 于是乎进入了我们前面描述的死循环。因为在执行get\_Length方法时, 还没有创建AsyncFileIO对象, 因此也就不会有工作线程来调用清理Busy标志的回调方法。因为没有想到CPU在执行get\_Length方法时飞将出去。这就是所谓的函数异常出口。图5包含了有关的代码和代表异常出口的示意箭头。P

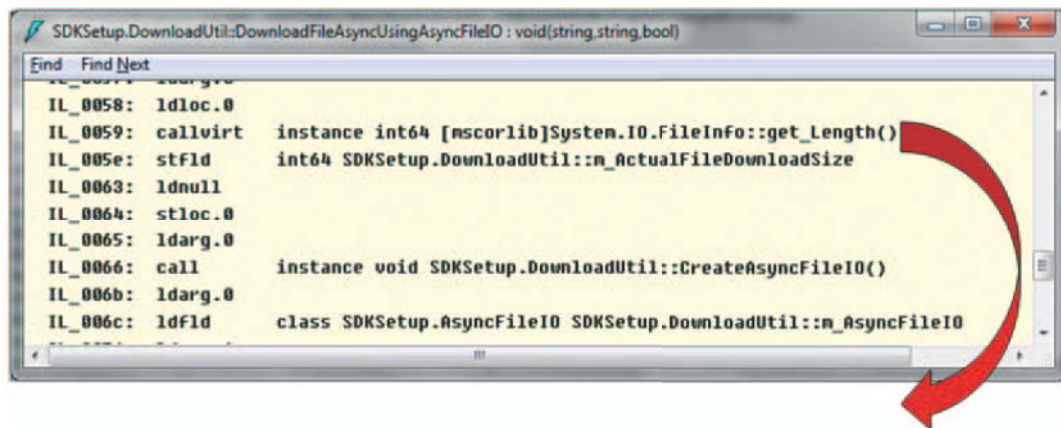


图5 看不见的异常出口

## 本期问题：

本期问题是：首先感谢来自北京的Thomson指出上一期文章中图4存在问题, 排版时不慎翻转错了方向, 我们深表歉意。上一期的问题是GetLastError() API是如何访问到TEB结构中的LastErrorValue字段的。读者只要在WinDBG中反汇编一下这个API便可以找到答案。比如在x86中, 这个API只包含三条指令, mov eax,dword ptr fs:[00000018h] mov eax,dword ptr [eax+34h] 和ret。其中FS寄存器用来指向特殊的线程信息块(一个很小的段), 偏移18的位置便是TEB。本期的问题是请大家阅读本期提到的ECMA-335文档, 找一下ldloc指令的含义和用法。

## 编者说明：

- 投稿信箱：  
contest@csdn.net
- 联系方式写在一个单独的TXT文件里, 包括以下几项：
  - 1) 姓名
  - 2) 工作单位或学校
  - 3) 电话联系方式
  - 4) 邮寄地址
  - 5) E-mail地址
- 解答提交时间, 最好早于当月15日。

# 深入Zend线程安全模型

文 / 张洋

本文首先解释了线程安全的概念及PHP中线程安全的背景，然后详细研究了PHP的线程安全机制ZTS及具体实现，最后探讨了Zend对于单线程和多线程环境的选择性编译问题。

## 线程安全

线程安全问题，一言以蔽之就是多线程环境下如何安全存取公共资源。我们知道，每个线程只拥有一个私有栈，共享所属进程的堆。在C中，当一个变量被声明在任何函数之外时，就成为一个全局变量，这时这个变量会被分配到进程的共享存储空间，不同线程都引用同一个地址空间，因此一个线程如果修改了这个变量，就会影响到全部线程。这看似为线程共享数据提供了便利，但是PHP往往是每个线程处理一个请求，因此希望每个线程拥有一个全局变量的副本，而不希望请求间相互干扰。

早期的PHP往往用于单线程环境，每个进程只启动一个线程，因此不存在线程安全问题。后来出现了多线程环境下使用PHP的场景，因此Zend引入了Zend线程安全机制（Zend Thread Safety，简称ZTS）用于保证线程的安全。

## ZTS的基本原理及实现

### 基本思想

说起来ZTS的基本思想是很直观的，不就是需要每个全局变量在每个线程都拥有一个副本吗？那我就提供这样的机制。

在多线程环境下，申请全局变量不再是简单声明一个变量，而是整个进程在堆上分配一块内存空间用作“线程全局变量池”，在进程启动时初始化这个内存池，每当有线程需要申请全局变量时，通过相应方法调用TSRM（Thread Safe

Resource Manager，ZTS的具体实现）并传递必要的参数（如变量大小等），TSRM负责在内存池中分配相应内存区块并将这块内存的引用标识返回，这样下次这个线程需要读写此变量时，就可以通过将唯一的引用标识传递给TSRM，TSRM将负责真正的读写操作。这样就实现了线程安全的全局变量。图1给出了ZTS原理的示意图。

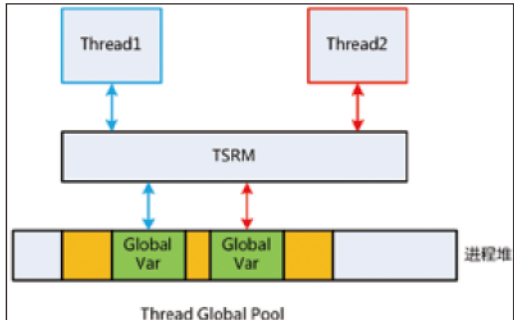


图1 ZTS原理的示意图

Thread1和Thread2同属一个进程，其中各自需要一个全局变量Global Var，TSRM为两者在线程全局内存池中（黄色部分）各自分配了一个区域，并且通过唯一的ID进行标识，这样两个线程就可以通过TSRM存取自己的变量而互不干扰。

下面通过具体的代码片段看一下Zend具体是如何实现这个机制的。这里我用的是PHP5.3.8的源码。

TSRM实现代码在PHP源码的“TSRM”目录下。

### 数据结构

TSRM中比较重要的数据结构有两个：tsrm\_tls\_entry和tsrm\_resource\_type。下面先看tsrm\_tls\_entry。



tsrm\_tls\_entry定义在TSRM/TSRM.c中:

```
typedef struct _tsrm_tls_entry tsrm_tls_entry;

struct _tsrm_tls_entry {
    void **storage;
    int count;
    THREAD_T thread_id;
    tsrm_tls_entry *next;
}
```

每个tsrm\_tls\_entry结构负责表示一个线程的所有全局变量资源,其中thread\_id存储线程ID, count记录全局变量数, next指向下一个节点。storage可以看做指针数组,其中每个元素是一个指向本节点代表线程的一个全局变量。最终各个线程的tsrm\_tls\_entry被组成一个链表结构,并将链表头指针赋值给一个全局静态变量tsrm\_tls\_table。注意,因为tsrm\_tls\_table是一个货真价实的全局变量,所以所有线程会共享这个变量,这就实现了线程间的内存管理一致性。tsrm\_tls\_entry和tsrm\_tls\_table结构的示意如图2。

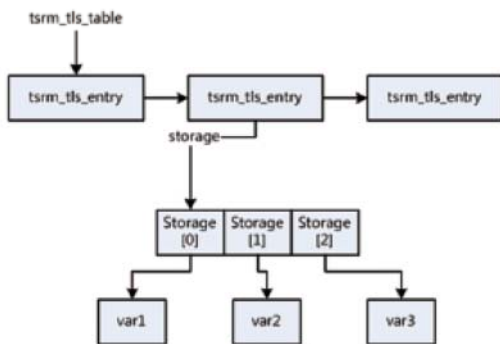


图2 tsrm\_tls\_entry和tsrm\_tls\_table结构示意图

tsrm\_resource\_type的内部结构相对简单一些:

```
typedef struct {
    size_t size;
    ts_allocate_ctor ctor;
    ts_allocate_dtor dtor;
    int done;
} tsrm_resource_type;
```

上文说过tsrm\_tls\_entry是以线程为单位的(每个线程一个节点),而tsrm\_resource\_type以资源(或者说全局变量)为单位,每次一个新的资源被分配时,就会创建一个tsrm\_resource\_type。所有tsrm\_resource\_type以数组(线性表)的方式组成tsrm\_resource\_table,其下标就是这个资源的ID。每个tsrm\_resource\_type存储了此资源的大小和构造、析构方法指针。某种程度上,tsrm\_resource\_table可以看做是一个哈希表, key是资源ID, value是tsrm\_resource\_type结构。

## 实现细节

这一小节分析TSRM一些算法的实现细节。因为整个TSRM涉及代码比较多,这里拣其中具有代表性的一个函数分析。

值得注意的是tsrm\_startup函数,这个函数在进程起始阶段被sapi调用,用于初始化TSRM的环境。由于tsrm\_startup略长,这里摘录出我认为应该注意的地方:

```
/* Startup TSRM (call once for the entire process) */
TSRM_API int tsrm_startup(int expected_threads, int expected_resources, int debug_level, char *debug_filename)
{
    /* code... */
    tsrm_tls_table_size = expected_threads;
    tsrm_tls_table = (tsrm_tls_entry *) calloc(tsrm_tls_table_size, sizeof(tsrm_tls_entry));
    if (!tsrm_tls_table) {
        TSRM_ERROR(TSRM_ERROR_LEVEL_ERROR, "Unable to allocate TLS table");
        return 0;
    }
    id_count=0;
    resource_types_table_size = expected_resources;
    resource_types_table = (tsrm_resource_type *) calloc(resource_types_table_size, sizeof(tsrm_resource_type));
    if (!resource_types_table) {
        TSRM_ERROR(TSRM_ERROR_LEVEL_ERROR, "Unable to allocate resource types table");
        free(tsrm_tls_table);
        tsrm_tls_table = NULL;
        return 0;
    }
    /* code... */
    return 1;
}
```

其实tsrm\_startup的主要任务就是初始化上文提到的两个数据结构。第一个比较有意思的是它的前两个参数: expected\_threads和expected\_resources。这两个参数由sapi传入,表示预计的线程数和资源数,可以看到tsrm\_startup会按照这两个参数预先分配空间(通过calloc)。因此TSRM会首先分配可容纳expected\_threads个线程和expected\_resources个资源的。要看各个sapi默认会传入什么,可以看各个sapi的源码(在sapi目录下),如图3。

可以看到比较常用的sapi如mod\_php5、php-fpm和cgi都是预分配一个线程和一个资源,这样是因

```
adserver/adserver.c: tsrm_startup(1, 1, 0, NULL);
apache/mod_php5.c: tsrm_startup(1, 1, 0, NULL);
apache/mod_php5.c: tsrm_startup(1, 1, 0, NULL);
apache2filter/sapi_apache2.c: tsrm_startup(1, 1, 0, NULL);
apache2handler/sapi_apache2.c: tsrm_startup(1, 1, 0, NULL);
apache_hooks/mod_php5.c: tsrm_startup(1, 1, 0, NULL);
apache_hooks/mod_php5.c: tsrm_startup(1, 1, 0, NULL);
caudium/caudium.c: tsrm_startup(1, 1, 0, NULL);
cgi/cgi_main.c: tsrm_startup(1, 1, 0, NULL);
cli/php_cli.c: tsrm_startup(1, 1, 0, NULL);
continuity/capi.c: tsrm_startup(128, 1, 0, NULL);
embed/php_embed.c: tsrm_startup(1, 1, 0, NULL);
fpm/fpm_main.c: tsrm_startup(1, 1, 0, NULL);
isapi/php5isapi.c: tsrm_startup(128, 1, TSRM_ERROR, NULL);
isapi/php5isapi.c: tsrm_startup(128, 1, TSRM_ERROR, NULL);
litespeed/lsapi_main.c: tsrm_startup(1, 1, 0, NULL);
milter/php_milter.c: tsrm_startup(1, 1, 0, NULL);
nsapi/nsapi.c: tsrm_startup(threads, 1, 0, NULL);
phttpd/phttpd.c: tsrm_startup(1, 1, 0, NULL);
pi3web/pi3web_sapi.c: tsrm_startup(1, 1, 0, NULL);
roxen/roxen.c: tsrm_startup(1, 1, 0, NULL);
thttpd/thttpd.c: tsrm_startup(1, 1, 0, NULL);
```

图3 各个sapi的源码

为不愿浪费内存空间，而且多数情况下PHP还是运行于单线程环境。

这里还可以看到一个id\_count变量，这个变量是一个全局静态变量，其作用就是通过自增产生资源ID，这个变量在这里被初始化为0。所以TSRM产生资源ID的方式非常简单：就是一个整型变量的自增。

ZTS与生命周期

这里需要简单提一下PHP的生命周期。

PHP的具体生命周期模式取决于sapi的实现，但一般都会有MINIT、RINIT、SCRIPT、RSHUTDOWN和MSHUTDOWN五个典型阶段，不同的只是各个阶段的执行次数不同。例如在CLI或CGI模式下，这五个阶段顺序执行一次，而在Apache或FastCGI模式下往往一个MINIT和MSHUTDOWN中间对应多个RINIT、SCRIPT、RSHUTDOWN。关于PHP生命周期的话题我回头写文单独研究，这里只是简单说一下。

MINIT和MSHUTDOWN是PHP Module的初始化和清理阶段，往往在进程起始后和结束前执行，在这两个阶段各个模块的MINIT和MSHUTDOWN方法会被调用。而RINIT、SCRIPT、RSHUTDOWN是每一次请求都会触发的一个小周期。在多线程模式中，PHP的生命周期如图4所示。

在这种模式下，进程启动后仅执行一次MINIT。

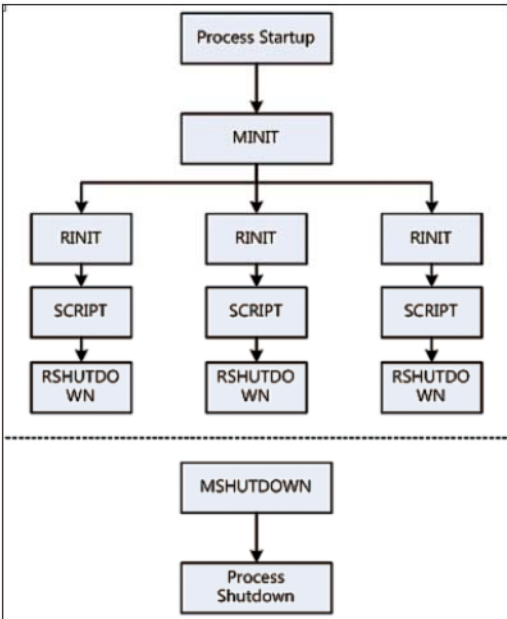


图4 PHP的生命周期图

之所以要强调这一点，是因为TSRM的全局变量资源分配就是在MINIT阶段完成的，后续阶段只获取而不会再请求新的全局变量，这就不难理解为什么在ts\_allocate\_id中每次id\_count加一需要遍历所有线程为每个线程分配相同的资源。到这里，终于看清TSRM分配线程全局变量的全貌：

进程启动后，在MINIT阶段启动TSRM（通过sapi调用tsrm\_startup），然后在遍历模块时调用每一个模块的MINIT方法，模块在MINIT中告知TSRM要申请多少全局变量及大小（通过ts\_allocate\_id），TSRM在内存池中分配并做好登记工作（tsrm\_tls\_table和resource\_types\_table），然后将凭证（资源ID）返回给模块，告诉模块以后拿着这个凭证来取你的全局变量。

ZTS在单线程和多线程环境的选择性编译

上文说过，很多情况下PHP还是被用于单线程环境，这时如果还是遵循上述行为，显然过于折腾。因为在单线程环境下不存在线程安全问题，全局变量只要简单声明使用就好，没必要搞那么一大堆动作。PHP的设计者考虑到的这一点，允许在编译时指定是否开启多线程支持，只有当在configure是指定enable-maintainer-zts选项或启用多线程sapi时，PHP才会编译线程安全的代码。具



体来说，当启用线程安全编译时，一个叫ZTS的常量被定义，PHP代码在每个与线程安全相关的地方通过#ifdef检查是否编译线程安全代码。

在探究相关细节前我先说一些自己的看法，对于ZTS多线程和单线程环境选择性编译设计上，我个人觉得是非常失败的。因为良好的设计应该隔离变化，换言之ZTS有义务将选择性编译相关的东西隔离起来，而不让其污染到模块的编写，这个机制对模块开发应该是透明的。但是ZTS的设计者仿佛生怕大家不知道有这个东西，让其完全污染了整个PHP，模块开发者不得不面对一堆奇奇怪怪的TSRM宏，着实让人非常不爽。所以下面我就带着悲愤的心情研究一下这块内容。

为了看看模块是如何实现选择性编译代码的，我们建立一个空的PHP扩展模块。到PHP源码的ext目录下执行如下命令：

```
./ext_skel --extname=zts_research
```

ext\_skel是一个脚手架程序，用于创建PHP扩展模块。此时会看到ext目录下多了个zts\_research目录。ext\_skel为什么生成了一个模块的架子，并附带了很多提示性注释。在这个目录下找到php\_zts\_research.h并打开，比较有趣是下一段代码：

```
/*
    Declare any global variables you may
    need between the BEGIN
    and END macros here:
    ZEND_BEGIN_MODULE_GLOBALS(zts_research)
        long global_value;
        char *global_string;
    ZEND_END_MODULE_GLOBALS(zts_research)
*/
```

很明显这里提示了定义全局变量的方法：用ZEND\_BEGIN\_MODULE\_GLOBALS和ZEND\_END\_MODULE\_GLOBALS两个宏包住所有全局变量。下面看一下这两个宏，这两个宏定义在Zend/zend\_API.h文件里：

```
#define ZEND_BEGIN_MODULE_GLOBALS(module_name) \
    typedef struct _zend_##module_name##_ \
    globals { \
#define ZEND_END_MODULE_GLOBALS(module_name) \
    } zend_##module_name##_globals;
```

原来这两个宏只是将一个模块的所有全局变量封装为一个结构体定义，名称为zend\_module\_name\_globals。

php\_zts\_research.h另外比较有意思的一处就是：

```
#ifdef ZTS
#define ZTS_RESEARCH_G(v) TSRMG(zts_research_globals_id, zend_zts_research_globals *, v)
#else
#define ZTS_RESEARCH_G(v) (zts_research_globals.v)
#endif
```

zts\_research\_globals是zts\_research模块全局变量结构的变量名称，类型为zend\_module\_name\_globals，在哪里定义的稍后会研究。这里ZTS\_RESEARCH\_G就是这个模块获取全局变量的宏，如果ZTS没有定义（非线程安全时），就直接从这个结构中获得相应字段，如果线程安全开启时，则使用TSRMG这个宏。

```
#define TSRMG(id, type, element) (((type) \
*((void ***) tsrm_ls))[TSRM_UNSHUFFLE_RSRC_ID(id)])->element)
```

这个宏就不具体细究了，因为实在太难懂了，基本思想就是使用上文提到的TSRM机制从线程全局变量池中获取对应的数据，其中tsrm\_ls可以看作是线程全局变量池的指针，获取变量的凭证就是资源ID。

---

ZTS有义务将选择性编译相关的东西隔离起来，而不让其污染到模块的编写，这个机制对模块开发应该是透明的。

---

看到这里可能还有点晕，例如zts\_research\_globals这个变量哪儿来的？zts\_research\_globals\_id又是哪儿来的？为了弄清这个问题，需要打开ext/zts\_research/zts\_research.c这个文件，其中有这样的代码：

```
/* If you declare any globals in php_zts_research.h uncomment this:
ZEND_DECLARE_MODULE_GLOBALS(zts_research)
*/
```

提示很清楚，如果在php\_zts\_research.h中定义了任何全局变量则将这段代码的注释消除，看来这个ZEND\_DECLARE\_MODULE\_GLOBALS宏就是关键了。然后在Zend/zend\_API中有这样的代码：

```
#ifdef ZTS

#define ZEND_DECLARE_MODULE_GLOBALS(module_name) \
    \
    ts_rsrc_id module_name##_globals_id; \
    /* code... */
#else
#define ZEND_DECLARE_MODULE_GLOBALS(module_name) \
```

# 寻找机遇 创造未来

## CSDN 热门职位全新推荐

### 上海爱数软件

上海爱数软件有限公司成立于 2003 年，是一家提供备份、容灾、存储整体解决方案的供应商，总部位于上海市。

**EISOO 爱数**  
创新思考 只为更好

#### 招聘职位:

- 销售经理及客户经理（全国）
- 存储、备份、云计算产品架构师、高级研发工程师（上海）

邮箱: zhang.junli@eisoo.com

更多详情, 请登录我公司官网 <http://www.eisoo.com/cn/>

### 阳光凯讯（北京）科技有限公司

成立于 2003 年，注册资金 2000 万，为北京市高新技术企业及软件企业。主营产品为应急移动通信系统，在软交换领域拥有国际先进、国内领先的核心技术。拥有多项软件著作权及专利权。



#### 现诚招:

C/C++ 软件工程师 4 人, Java 软件工程师 3 人, 测试工程师 3 人, LTE 研发、DSP、FPGA、射频、硬件各 1 人。

公司地址: 北京丰台科学城航丰路 9 号颐安鑫鼎 6 层

网址: [www.sunkaisens.com](http://www.sunkaisens.com) e-mail: [hr@sunkaisens.com](mailto:hr@sunkaisens.com)

### 网易有道

网易有道是网易自主研发的中文搜索引擎，致力于为互联网用户提供更快更好的中文搜索服务。目前有道已推出的产品包括网页搜索、图片搜索、新闻搜索、购物搜索、视频搜索、音乐搜索、海量词典、桌面词典、工具栏和有道笔记。



#### 网易有道常年热招职位:

- 研发工程师 ■ web 开发工程师 ■ 测试工程师

欢迎投递 [wangjian@rd.netease.com](mailto:wangjian@rd.netease.com)

```
zend_#module_name##_globals module_
name##_globals;
/* code... */
#endif
```

当线程安全开启时，这里实际定义了一个整形的资源ID（ts\_rsrc\_id 被typedef定义为int），而当线程安全不开启时，则直接定义一个结构体。在这个模块中分别对应zts\_research\_globals\_id和zts\_research\_globals。

到这里思路基本理顺了：如果ZTS没有被启用，则直接声明一个全局变量结构体，并直接通过存取其字段实现全局变量存取；如果ZTS开启，则定义一个整形变量作为资源ID，然后通过ts\_allocate\_id函数向TSRM申请一块内存放置结构体（需要程序员手工在MINIT函数中实现，脚手架生成的程序中没有），并通过TSRM存取数据。

### 总结

本文研究了PHP和Zend的线程安全模型，个人感觉Zend内核中ZTS的实现巧妙但不够优雅，但目前在开发PHP模块时总免不了常与之打交道。这块内容相对偏门，几乎没有资料对ZTS和TSRM进行详细解释，但是透彻了解ZTS机制对于在PHP模块开发中正确合理使用全局变量是很重要的。希望本文对读者有所帮助。P



张洋

花名夜澜，淘宝开发工程师。目前供职于淘宝量子统计团队，主要从事PHP、Lua、MySQL、Nginx及Web前端的开发工作。

责任编辑: 高松 (gaosong@csdn.net)



# 云计算时代的多核开发

文 / 陈冠诚

云计算时代的多核开发会是一幅什么样的场景？这两大趋势彼此会有什么样的影响？本文就这些问题做了探讨。

云计算和多核这两大趋势正对软件开发者产生重大影响。一方面，近几年多核逐渐成为主流，随着CPU核心频率提升越来越难，处理器厂商选择了更加容易实现的多核方案来继续提升硬件的性能。进入后PC时代，移动处理器也同样面临着性能的提升与功耗的控制这两大挑战，为了满足提升性能与控制功耗的需求，多核也正成为处理器发展的方向。另一方面，云计算渐渐成为软件开发的大势。云计算生态系统中最主要的设备是“端”和“云”。端包括移动设备（智能手机、Pad等）和传统PC，尤其是前者；云指的是由高性能服务器组成的大规模集群，它们向端设备提供各种服务支持。不妨先回顾一下在大型机和PC机时代软件开发的历史。

## 多核开发将更加容易

在大型机时代，计算机非常昂贵，用户需要分时共享同一台大型机。计算资源的稀缺使得那时的软件开发者必须高效利用每一个处理器时钟周期，因此他们大都使用汇编、C等非常底层的语言来开发软件，而算法效率是他们最关心的问题。在之后的几十年中，计算机硬件变得越来越廉价，软件开发者越来越不需要关心软件性能。以主流的互联网应用为例，现在开发大量使用成熟的框架来帮助自动生成大量的代码。就拿Django这个流行的Web开发框架来说，它的设计原则是“focuses on automating as much as possible and adhering to the DRY principle: Don't Repeat Yourself”。开发者最核心的目标已经变成了如何用最少的代码、最快的速度将自己的点子转为成可用的软件产品并推向市场。“市场投放时间”

已经取代“处理器时钟周期”成为软件开发的关键指标。

在过去的几十年里，正是因为硬件一直在按照摩尔定律稳步发展，所以开发者不再需要时刻关注软件性能，而是将注意力转移到更为重要的开发效率上，这点在近十年来Java、Python和Ruby等高级语言的兴起上就可见一斑。多核的出现，将硬件细节再一次暴露在程序员的面前。想利用好多核，程序员必须手动处理同步、死锁和数据竞跑等疑难问题，这极大降低了软件开发效率。现有的生产工具（多核开发框架、开发工具）远不能满足生产力（软件开发效率）的发展需要，还有很大的发展空间。可以预见，不久的将来更简单易用的多核开发框架将不断涌现，在多核上并行编程将变得越来越容易。那放在云计算的大背景下，多核开发又会有怎样的发展呢？先来看看在“云”和“端”上的多核发展趋势。

## “云”和“端”的多核趋势

据IDC预测，以智能手机和Pad为代表的移动设备在2013年将达到3.9亿台的出货量；相对的，传统PC机、笔记本和服务器的出货量预计为4.4亿。移动设备的日益流行将让更多开发者转向移动平台。与此同时，云将为端设备提供更多服务支撑。那么云和端上的多核将如何发展呢？

如表1所示，从2012年开始“双核”手机/平板将成为主流。因为受到功耗的限制，移动设备上的

	Number of cores				
	(45 nm) Current	(32nm) 2012	(24 nm) 2014	(19 nm) 2016	(15 nm) 2018
Mobile-devices	1	2	3-4	5-6	9-10
servers	8	16	28	44	72

表1 多核将成为主流

处理器核数并不会迅速增长。实际上，移动设备将会越来越多依赖专用硬件加速器来提供高性能、低功耗的解决方案。GPU是个很好的例子。在手机和平板上观看高清电影、玩高分辨游戏时，我们可以依靠专用图形处理器来进行图像渲染、高清解码等操作，这种解决方案相比于使用更多的通用处理器核数来说，能提供更高的性能功耗比。从开发者的角度来讲，产品设计、用户体验才是现阶段移动开发者最关注的问题，利用并行编程的方式提升移动应用的性能，短期内还不是最主要的关注点。不可否认，越来越多的移动应用将通过并行化的方式提供更绚丽的3D渲染、更流畅的用户体验以及更丰富的特效（尤其是游戏类应用）。

---

多核和并行编程的普及，使得处于云端的开发者不太需要担心如何用并行编程来解决面临的问题。

---

与此同时，云端服务器的处理器核数将继续以每18个月翻一番的速度增长。在多核之前，软件开发者无需担心软件的性能，他们唯一要做的就是“等”：等到下一代处理器出现，软件对性能的需要就能得到满足。这个免费午餐在多核到来之后不复存在：单纯靠增加处理器的核数并不能提升单线程程序的性能。换言之，我们必须通过并行的方式来提升“串行”应用的性能。但是如果我们所关心的问题不再是如何提升单线程的性能，而是如何利用更多的核来处理已经并行化的应用，那么核数的增加不就能继续“免费”提升此类应用的性能吗？从这个角度来看，云端的应用与多核有点天生一对的意味。举例来说，以Hadoop为基础的大规模数据处理通过并行执行Map和Reduce来有效处理海量数据。这种数据并行模式关心的不再是单个Mapper或者Reducer的性能，而是所有Mapper和Reducer的吞吐量。如果需要处理的数据增加了，我们一般只需要增加更多的机器（即更多处理器核数）就能达到所需的性能。

谈到并行计算时，必须区分好两种完全不同的应用：并行与并发。并行是指两个或多个Task同时执行，以完成同一个计算任务，例如使用两个线程来并行完成矩阵乘运算。并发是指两个或多个

Task同时执行，但是彼此相互独立、分别在完成不同的计算（这里的Task不仅仅局限于线程，它也可以代表纤程、进程等）。对云计算来说，云端所需要处理的请求大都是并发任务，因为不同的终端请求彼此大都是相互独立的。数千用户同时使用Google Docs编辑文件，服务器端需要处理的就是数千个并发请求，这些独立请求能自然地把服务器上的多核利用好。在云计算的大背景下，大量存在的并发应用能天然利用好云端的多核，通过并行方式来利用好多核并不是那么的重要。

## 人人都是并程序员？

多核出现之初，许多业界人士都惊呼“狼来了”，人人都需要掌握并行编程。殊不知并行编程这项技术早在二三十年前就已经存在了，只不过当时大都是搞高性能计算的一小群人会并行编程，而随着多核的普及，并行编程的神秘面纱也逐渐向大众展开。幸运的是，在云计算的大图下，多核的应用场景以及与高性能计算领域大不相同。

高性能领域关心的主要问题是如何用更多的处理器核心来更快的完成同一个任务，例如天气预测、地震模拟等。而在云计算领域，我们面临的主要难题是如何满足众多端设备的并发请求，这些请求彼此大都独立，处于云端的开发者已经不太需要担心如何用并行编程来解决他们所面临的问题。

Google趋势中“云计算”这个关键词的热度一直都处在上升趋势中，而“多核（multicore）”的热度一直都比较平稳。随着移动互联网的兴起，Android和iOS开发的热度也已经超过了多核。并不是所有的程序员都需要关心如何进行并行编程。在云计算的大背景下，并发应用能与多核很容易地结合在一起，将云端的多核利用好。P



陈冠诚

IBM中国研究院研究员，瑞典查尔姆斯理工大学（Chalmers）硕士，主要研究方向为并行计算、计算机体系结构及分布式系统。个人博客为并行实验室（<http://parallellabs.com>）。

责任编辑：高松（[gaosong@csdn.net](mailto:gaosong@csdn.net)）



# 新书上架

本月新书



研究之美

作者: Donald E. Knuth

译者: 高博

本书是作者旅居挪威时撰写的计算机科学的全新基本数学结构情景小品。主人公为一对青年男女, 全书以对白形式展开对于该种全新结构的发现、构造, 以及在算法和效率的应用之研究。



Ruby元编程

作者: Paolo Perrotta

译者: 廖志刚 陈睿杰

以案例形式介绍Ruby编程技巧。通过分析案例、讲解例题和回顾Ruby代码库的实现细节, 展示了Ruby编程的优势和特有的解决问题的方式, 并列出发挥优势的技巧和常用编程设计模式。



Oracle DBA手记3: 数据库性能优化与内部原理解析

作者: 盖国强 杨廷琨 等

本书的主要内容是原理分析、内部实践和故障解决, 将Oracle数据库深层技术展示给读者。帮助读者加深对Oracle技术的认知和理解, 并且将这些技术应用到实践中去。



布道之道

作者: Ryan, T

译者: 李纁 李松峰

本书将怀疑新技术的人分成了七种类型: 孤陋寡闻型、随波逐流型、百般挑剔型、激情燃尽型、时间紧迫型、发号施令型和不可理喻型, 进而给出在推广新技术时有效的应对技巧和说服策略。



Android应用开发实战

作者: 李宁

本书的实践部分以一个功能完善的微博客户端开发贯穿始终, 并作了迭代式讲解, 理清应用开发的完整流程和实现细节; 理论部分阐述了Android应用开发所需具备的高级知识和常用技巧。

推荐图书



奇点临近

作者: Ray Kurzweil

译者: 李庆诚 董振华 田源

出版社: 机械工业出版社

“奇点”一词来源于数学的 $Y=1/X$ 函数曲线上 $X=0$ 的一点, 它是一个未被定义的点。在物理学上, 奇点被拿来作为“大爆炸理论”中的起点, 也因此, 它带给人们无限的遐想。在作者看来, 奇点预示着科技发展的下一个阶段: 人和机器的联合, 即嵌入我们大脑的知识和技巧将与我们创造的容量更大、速度更快、知识分享能力更强的职能相结合。这种融合便是奇点的本质。

在这个时代, 我们的智能会逐渐非生物化, 智能程度将远远高于今天的智能——一个新的文明正在冉冉升起, 它将让我们超越人类的生物局限, 大大加强我们的创造力。在这个新世界中, 人类与机器、现实与虚拟的区别将变得模糊, 我们可以任意地装扮成不同的身体, 扮演一系列角色。它所带来的实际效果包括: 人类将不再衰老, 疾病将被治愈, 环境污染将会结束, 世界性的贫困、饥饿问题都会得到解决。纳米科技通过使用廉价的处理器, 几乎能够创造任意的实际产品, 甚至可以最终解决起死回生的问题。社会和哲学各个分支的变革将是意义深远的, 同时也暗含着巨大的威胁。书中阐述了对人类未来发展进程的本质乐观主义观点, 基于此, 也提供了一个崭新的视角: 即将到来的时代, 既是数百年来科技、创意的顶点, 也是对人类终极命运真挚的、鼓舞人心的愿景。

Kurzweil认为发明的意义在于能够为技术和市场力提供能量, 以构建远不同于原来世界的新世界。他开始研究人工智能, 并坚信在21世纪前半叶, 人工智能可以媲美人类祖先的智能, 改变世界的思想力量其本身也在加速。书中六个纪元的划分非常奇特而又富于哲理: 宇宙的唤醒是个神圣的命题, 大脑的模拟、计算能耗、人与机器的相互融合等都是一个个令人惊叹的结论, 其结论如此的积极、自信和奇特。书中前两章运用了归纳式推理总结, 后几章又大量运用了演绎式推理预测诠释了“回溯有多远, 预见就有多远”的道理; 通过追溯、分析以往的科学发展趋势和当今科技的现状, 演绎并预测未来。书中每章开始都有名人的警世之语: 通过智者的眼睛去审视自然、科学以及我们生存的世界, 从而不断强化思想的力量; 章尾是与未来的对话, 是一种思想的博弈: 通过设想中的未来去理解当今的技术发展和进化中的人类。书中集中体现了西方科学注重演绎的思维方式, 即分科放大的纵向思维回归; 不同于东方哲学注重归纳的横向思维方式; 书中以东方的归纳思维为基础, 以西方的演绎思维导出结论, 两者恰如“T”字形中的“横”与“竖”, 无意之中完成了一次东西方思想的对话。

奇点概念一提出, 便引发了无数的衍生物, 其中包括“奇点主义”。奇点已经成为一种思潮, 在世界范围内具有广泛影响, 关于它的争论一天没有停息过, 拥护者将奇点升级为奇点主义, 从哲学、科学、技术和艺术等各个方面建立奇点。2009年2月, Google和NSNA联合建立了奇点大学, 旨在解决“人类面临的重大挑战”。作者Kurzweil还自编、自导、自演了一部《奇点临近》的电影, 从艺术的角度说明“奇点”。



### 编程大师访谈录

作者: Lammers, S.

译者: 李琳骁 张菁 吴咏炜

出版社: 人民邮电出版社

本书副标题为“塑造计算机工业19位编程大师访谈”，内容取自作者Susan Lammers1986年在微软工作时，对一系列软件行业领袖的采访。

个人电脑发展的初期犹如西部小说里的荒野。拓荒者、淘金者和冒险家们汇聚于此，他们没有称手的工具，没有足够的资金，缺乏经验，不知前方等待他们的将是怎样的艰险与危机。他们拥有的只是勤劳的手脚、聪慧的大脑和理想主义的热情。书中介绍的19位编程领域的大师扮演的正是这样的角色。

那是软件行业刚刚起步的时代，他们最先接触到那个时代最先进的电脑。虽然也许只有4KB的内存、40位的浮点运算和20位操作，但他们往往花掉一整个下午，在电脑上演练自己的编程技艺。技术还很原始，但对他们来说编程迈出最重要的一步是不受羁绊的想象力。原始的工具，激起了他们改变人类社会的雄心壮志。那个时代没有数不清的类库框架，他们只能啃着B.J. Plauger的大部头；也没有巨细靡遗的操作系统，他们就回去仔细研读Lion的经典。虽然他们最初的成果只是一个非常简单的类FORTRAN语言的编译器，或者是只能上下左右移动的巴比特游戏，但他们却在一直创造着奇迹。作为程序员，当我们遥望他们，就像望向深远的星空，那一段段动人的往事令人神往而感动。

在这本书里，我们能看到，杰夫·拉斯金如何让Macintosh成为个人电脑的不朽传奇？丹·布兰克林如何灵光闪现发明了Spreadsheet？约翰·沃诺克怎么成为PostScript语言的传奇开发者，C.韦恩·莱特莱夫在何种艰难处境下发明了dBASE，岩谷彻如何从食物中寻找灵感创造了Pac Man，商业天才比尔·盖茨其实也是开发了BASIC语言的黑客高手，巴特勒·兰普森发明所见即所得编辑器和Ethernet时，经历了怎样的顿悟？雷·奥奇从哪里得到关于群件的灵感？乔纳森·萨奇如何顶住巨大压力最终成为Lotus 1-2-3的联合作者等。

编程是什么？人们为此不停争执，有人认为这是一门科学，有人称其为艺术，也有人觉得不过是一项技艺。虽然大师们的解释也各不相同，但他们观点的交集似乎也是科学、艺术和技艺的交集。

每篇访谈都可看做独特的成功故事，19篇合在一起就生动地刻画了20世纪80年代中期整个PC产业的蓬勃兴起。你可以从中看到莲花软件公司、苹果电脑公司、施乐帕洛阿尔托研究中心和微软公司等著名软件公司的草创和发展演变过程。

正如Peter Norvig在书评里所说，理解编程，除了学习黑客们铿锵的代码外，也许还要了解他们丰富的内心。虽说高手不可模仿，因为“学问跟他整个的性情陶融为一片，不仅有丰富的数量，还添上个别的性质；每一个琐细的事实，都在他的心血里沉浸滋养，长了神经和脉络，是你所学不会、学不到的”，但我们无疑可以受到不小的启发。如果能被他们打动，分享同样的激情，也算一桩美事。

## 全球排行榜

### Amazon

- 01 Steve Jobs
- 02 Head First Java, 2nd Edition
- 03 Head First HTML5 Programming: Building Web Apps with JavaScript
- 04 JavaScript: The Good Parts
- 05 Arduino: A Quick Start Guide
- 06 Programming Android
- 07 Objective-C Programming: The Big Nerd Ranch Guide
- 08 JavaScript: The Definitive Guide: Activate Your Web Pages
- 09 Cracking the Coding Interview: 150 Programming Questions and Solutions
- 10 C Programming Language

### 天珑书局（中国台湾）

- 01 Eclipse完全攻略：从基础JAVA到PDE外挂开发
- 02 Android技术内幕：探索Android核心原理与系统开发
- 03 Google Android应用程序开发实战，3/e
- 04 一定要学会的HTML5+CSS3网页设计实作应用
- 05 深入浅出Servlets与JSP，2/e
- 06 前进Android Market！Google Android SDK实战演练
- 07 Google Android SDK开发范例大全，3/e
- 08 ASP.NET 4.0专题实务：使用C#
- 09 鸟哥的Linux私房菜：基础学习篇，3/e
- 10 Google! Android 3手机应用程序设计入门，4/e

### 第二书店

- 01 Java编程思想（第4版）
- 02 构建高可用Linux服务器
- 03 布道之道：引领团队拥抱技术创新
- 04 C++反汇编与逆向分析技术揭秘
- 05 持续交付：发布可靠软件的系统方法
- 06 设计模式：可复用面向对象软件的基础
- 07 算法导论（原书第2版）
- 08 黑客（计算机革命的英雄）
- 09 PostgreSQL数据库内核分析
- 10 PHP和MySQL Web开发（原书第4版）



# GEEK产品



## ◉ G-Form Extreme Portfolio

G-Form原来专注于开发能给运动员提供保护的产品，目前它将业务范围扩大，开始为平板设备生产保护套。这款iPad保护套不仅防水，还具有超强的防摔功能，外壳在撞击过程中迅速变硬，吸收最高90%的因撞击产生的冲击力。实验测试显示，将带有该保护套的iPad从390米高空扔下，仍然毫发无损。



## ◉ Celluon Magic Cube

由韩国Celluon公司推出的无线激光投影虚拟键盘Celluon Magic Cube穿越了实体与虚拟的界限，通过内置的感应器模块可定位操作者手指的位置，追踪手指的移动轨迹，每分钟可接收约400次敲击按键的识别。该虚拟键盘不仅可以应用于桌面操作系统，还能配合iPad/iPhone以及Android2.2等移动设备和手机使用，支持USB和蓝牙。





### ◆ Belkin Wireless Dual-Band Travel Router

想要保持网络随时在线的状态？最重要的是拥有一个无线热点能让移动设备随时接入网络。Belkin的这款旅行用无线路由器只有7cmx10cm大小，支持WPS一键快速配置和PPTP IPSec VPN，让旅途通信更便捷。

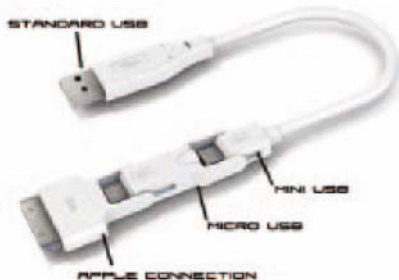
### ◆ Magic Cable Trio

这是由设计公司Innergie推出的一款三合一数据线。数据线共有三个部分：长达20cm的Mini USB数据线、Micro USB转接口及Apple 40 pin的转接口。Mini USB可供Android手机、MP3、数码相机等使用，Micro USB适用于黑莓、eBook，Apple 40 pin转接口可用于iPad、iPhone、iPod设备。三根数据线可连接在一起合为一条，能解决几乎所有数码设备的数据线问题。



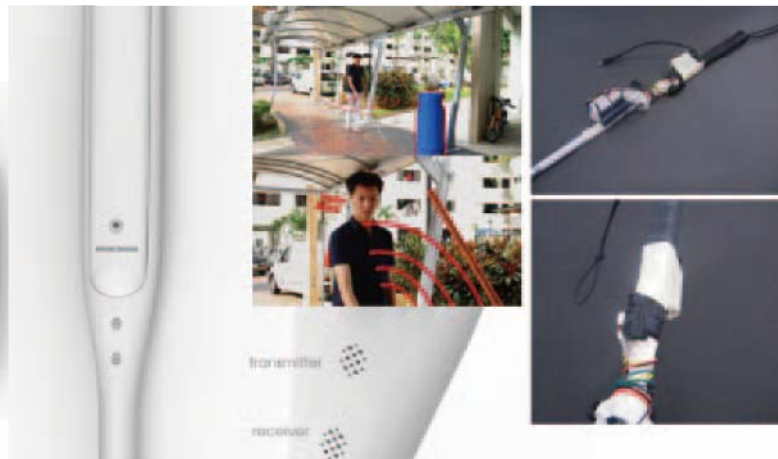
### ◆ 透明质多点触摸键盘鼠标

毋庸多言，这个近乎科幻的外观已经足以震撼你的眼球！这是一款由创业项目网Kickstarter开发的创意产品，采用亚克力透明材质的多点触摸鼠标和键盘，内置一个嵌入式摄像头，通过FTIR（受抑全内反射）技术实现多点触摸。该产品的设计者Jason Giddings还计划将产品所有软件都通过开源方式实现。



### ◆ Blindspot导盲杖

这是由一位漫画爱好者设计，并获得英国詹姆斯戴森设计比赛亚军的作品。该智能型导盲杖包含一套由导盲杖、内置手机、超声波传感器、蓝牙耳机、触控操作滚轮组成的智能系统：内置手机可让视障人士在紧急时刻与朋友通话；超声波传感器能够让视障人士避开悬挂着的障碍物；原本以音效传达的导盲指令改成直接以滚轮设定正确方向，直接把视障人士带向正确的方向。





Mac OS X 背后的故事 (八)

# 三好学生Chris Lattner的LLVM编译工具链

文 / 王越

2011年12月3日, LLVM 3.0正式版发布, 完整支持所有ISO C++标准和大部分C++ 0x的新特性, 这对于一个全新项目来说非常不易。

## 开发者的惊愕

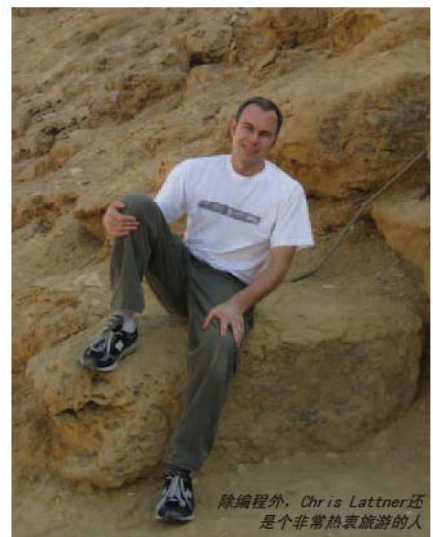
在2011年WWDC(苹果全球开发者大会)的一场与Objective-C相关的讲座上, 开发者的人生观被颠覆了。

作为一个开发者, 管理好自己程序所使用的内存是天经地义的事, 好比人们在遛狗时必须清理狗的排泄物一样(美国随处可见“Clean up after your dogs”的标志)。在本科阶段上C语言的课程时, 教授们会向学生反复强调: 如果使用malloc函数申请了一块内存, 使用完后必须再使用free函数把申请的内存还给系统——如果不还, 会造成“内存泄漏”的结果。这对于Hello World可能还不算严重, 但对于庞大的程序或是长时间运行的服务器程序, 泄内存是致命的。如果没记住, 自己还清理了两次, 造成的结果则严重得多——直接导致程序崩溃。

Objective-C有类似malloc/free的对子, 叫alloc/dealloc, 这种原始的方式如同管理C内存一样困难。所以Objective-C中的内存管理又增加了“引用计数”的方法, 也就是如果一个物件被别的物件引用一次, 则引用计数加一; 如果不再被

该物件引用, 则引用计数减一; 当引用计数减至零时, 系统则自动清掉该物件所占的内存。具体来说, 如果我们有一个字符串, 当建立时, 需要使用alloc方法来申请内存, 引用计数则变成了一; 然后被其他物件引用时, 需要用retain方法去增加它的引用计数, 变成二。当它和刚才引用的物件脱离关联时, 需使release方法减少引用计数, 又变回了一; 最后, 使用完这个字符串时, 再用release方法减少其引用计数, 这时, 运行库发现其引用计数变为零了, 则回收走它的内存。这是手动的方式。

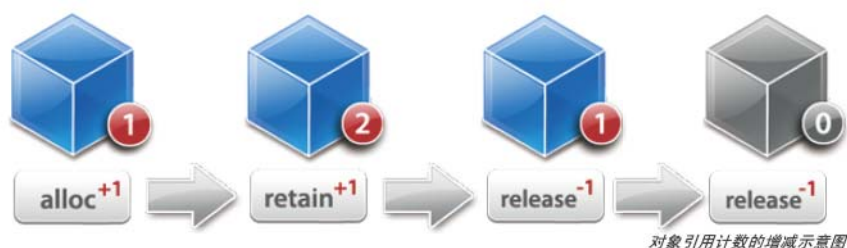
这种方式自然很麻烦, 所以又设计出一种叫做autorelease的机制(不是类似Java的自动垃圾回收)。在Objective-C中, 设计了一个叫做NSAutoreleasePool的池, 当开发者需要完成一个任务时(比如每开启一个线程, 或者开始一个函数), 可以手动创立一个这样的池子, 然后通过显式申明把物件扔进自动回收池中。NSAutoreleasePool内有一个数组来保存声明为autorelease的所有对象。如果一个对象声明为autorelease, 则会自动加到池子里。如果完成了一个任务(结束线程了, 或者退出那个函数), 开发者则需对这个池子发送一个



除编程外, Chris Lattner还是个非常热衷旅游的人

drain消息。这时, NSAutoreleasePool会对池子中所有的物件发送release消息, 把它们的引用计数都减一——这就好比游泳池关门时通知所有客人都“滚蛋”一样。所以开发者无须显式声明release, 所有的物件也会在池子清空时自动呼叫release函数, 如果引用计数变成零了, 系统才回收那块内存。所以这是个半自动、半手动的方式。

Objective-C的这种方式虽然比起C来进了一大步, 我刚才花了几分钟就和读者讲明白了。只要遵守上面这两个简单的规则, 就可以保证不犯任何错误。但这和后来的Java自动垃圾回收相比则是非



常繁琐的，哪怕是再熟练的开发者，一不小心就会弄错。而且，哪怕很简单的代码，比如物件的getter/setter函数，都需要用户写上一堆的代码来管理接收来的物件的内存。

经典教材《Cocoa Programming for Mac OS X》用整整一章节的篇幅，来讲解Objective-C中内存管理相关的内容，但初学者们看得还是一头雾水。所以，在2007年10.5发布时，Objective-C做出了有史以来最大的更新，最大的亮点是它的运行库libobjc 2.0正式支持自动垃圾回收，也就是由运行库在运行时随时侦测哪些物件需要被释放。听上去很不错，可惜使用这个技术的项目却少之又少。原因很简单，使用这个特性，会有很大的性能损失，使Objective-C的内存管理效率低得和Java一样，而且一旦有一个模块启用了这个特性，这个进程中所有的地方都要启用这个特性——因此如果写了一个使用垃圾回收的库，那所有引用库的程序就都得被迫使用垃圾回收。所以苹果自己也不使用这项技术，大量的第三方库也不使用它。

这个问题随苹果在移动市场的一炮走红而变得更加严峻。不过这次，苹果和与会的开发者讲，他们找到了一个解决问题的终极方法，这个方法把从世界各地专程赶来聆听圣谕的开发者惊得目瞪口呆——你不用写任何内存管理代码，也不需要使用自动垃圾回收。因为我们的编译器已经学会了上面所介绍的内存管理规则，会自动在编译程序时把这些代码插进去。

其实，这个编译器，一直是苹果公开的秘密——LLVM。说它公开，是因为它自始至终都是一个开源项目；而秘密，则是因为它从来没公开在WWDC的Keynote演讲上亮相过。

一直关注这系列连载的读者一定还记得，在第二篇《Linus Torvalds的短视》介绍苹果和GPL社区的不合时，提到过“自以为是但代码又写得差的开源项目，苹果事后也遇到不少，比如GCC编译器项目组。虽然大把钞票扔进去，在先期能够解决一些问题，但时间长了这群人总和苹果过不去，并以自己在开源世界的地位恫吓之，最终苹果由于受不了这些项目组的态度、协议、代码质量，觉得还不如自己造轮子来得方便。”LLVM则是苹果造的这个轮子，它的目的是完全替代掉GCC那条编译链。它的主要作者，则是现在就职于苹果的Chris Lattner。

### 编译器高材生Chris Lattner

2000年，本科毕业的Chris Lattner像中国多数大学生一样，按部就班地考了GRE，最终前往UIUC（伊利诺伊大学厄巴纳香槟分校），开始了艰苦读计算机硕士和博士的生涯。在这阶段，他不仅周游美国各大景点【注：<http://photos.nondot.org/>】，更是努力学习科学文化知识，翻烂了“龙书”（《Compilers: Principles, Techniques, and Tools》），以4.0的满分成了GPA牛人，以及不断地研究探索关于编译器的未知领域，发表了一篇又一篇的论文，是中国传统观

念里的“三好学生”。他的硕士毕业论文【注：LLVM: An Infrastructure for Multi-Stage Optimization】提出了一套完整的在编译时、链接时、运行时，甚至是在闲置时优化程序的编译思想，直接奠定了LLVM的基础。

LLVM在他念博士时更加成熟，使用GCC作为前端来对用户程序进行语义分析产生IF（Intermediate Format），然后LLVM使用分析结果完成代码优化和生成。这项研究让他在2005年毕业时，成为小有名气的编译器专家，他也因此早早地被苹果相中，成为其编译器项目的骨干。

苹果相中Chris Lattner主要是看中LLVM能摆脱GCC束缚。苹果（包括中后期的NeXT）一直使用GCC作为官方的编译器。GCC作为开源世界的编译器标准一直做得不错，但苹果对编译工具会提出更高的要求。

一方面，是苹果对Objective-C语言（甚至后来对C语言）新增很多特性，但GCC开发者并不买苹果的账——不给实现，因此索性后来两者分成两条分支分别开发，这也造成苹果的编译器版本远落后于GCC的官方版本。另一方面，GCC的代码耦合度太高，不好独立，而且越是后期的版本，代码质量越差，但苹果想做的很多功能（比如更好的IDE支持）需要模块化的方式来调用GCC，但GCC一直不给做。甚至最近，《GCC运行环境豁免条款（英文版）》从根本上限制了LLVM-GCC的开发。所以，这种不和让苹果一直在寻找一个高效的、模块化的、协议更放松的开源替代品，Chris Lattner的LLVM显然是一个很棒的选择。刚进入苹果，Chris Lattner就大展身手：首先在OpenGL小组做代码优化，把LLVM运行时的编译架在OpenGL栈上，这样OpenGL栈能够产出更高效率的图形代码。如果显卡足够高级，这些代码



会直接扔入GPU执行。但对于一些不支持全部OpenGL特性的显卡（比如当时的Intel GMA卡），LLVM则能够把这些指令优化成高效的CPU指令，使程序依然能够正常运行。这个强大的OpenGL实现被用在了后来发布的Mac OS X 10.5上。同时，LLVM的链接优化被直接加入苹果的代码链接器上，而LLVM-GCC也被同步到使用GCC4代码。

正像名字所写的那样，Clang只支持C，C++和Objective-C三种C家族语言。2007年开始开发，C编译器最早完成，而由于Objective-C相对简单，只是C语言的一个简单扩展，很多情况下甚至可以等价地改写为C语言对Objective-C运行库的函数调用，因此在2009年时，已经完全可以用于生产环境。C++的支持也热火朝天地进行着。

比如这个程序：

```
#include <inttypes.h>
int64 x;
```

GCC一样给出乱码似的出错提示：

```
t.c:2: error: expected '=', ',', ';',
'asm' or '__attribute__' before 'x'
```

而优雅的Clang则用彩色的提示告诉你是不是拼错了，并给出可能的变量名：

```
t.c:2:1: error: unknown type name
'int64'; did you mean 'int64_t'?
int64 x;~~~~~int64_t
```

【更多的例子可以参考<http://blog.llvm.org/2010/04/amazing-feats-of-clang-error-recovery.html>】

而同时又因为Clang是高度模块化的一个前端，很容易实现代码的高度重用。所以比如Xcode 4.0的集成编程环境就使用Clang的模块来实现代码的自动加亮、代码出错的提示和自动的代码补全。开发者使用Xcode 4.0以后的版本，可以极大地提高编程效率，尽可能地降低编译错误的发生率。

支持C++也是Clang的一项重要使命。C++是一门非常复杂的语言，大多编译器（如GCC、MSVC）用了10年甚至20多年来完善对C++的支持，但效果依然不很理想。Clang的C++支持却一直如火如荼地展开着。2010年2月4日，Clang已经成熟到能自举（即使用Clang编译Clang，到我发稿时，LLVM 3.0发布已完整支持所有ISO C++标准，以及大部分C++ 0x的新特性。

这对于一个短短几年的全新项目来说是非常不易的。得益于本身健壮的架构和苹果的大力支持，Clang越来越全能，从FreeBSD到Linux Kernel，从Boost到Java虚拟机，Clang支持的项目越来越多。

苹果的Mac OS X以及iOS也成了Clang和LLVM的主要试验场——10.6时代，很多需要高效运行的程序比如OpenSSL和Hotspot就由LLVM-GCC编译来加速的。而10.6时代的Xcode 3.2诸多图形界面开

---

LLVM真正的发迹，得等到Mac OS X 10.6 Snow Leopard登上舞台。可以说，Snow Leopard的新功能，完全得益于LLVM的技术。而这一个版本，也是将LLVM推向真正成熟的重大机遇。

---

关于Snow Leopard的三项主推技术（64位支持、OpenCL，以及Grand Central Dispatch）的细节，我们会在下次有整整一期篇幅仔细讨论，这次只是点到为止——我们告诉读者，这些技术，不但需要语言层面的支持（比如Grand Central Dispatch所用到的“代码块”语法，这被很多人看做是带lambda的C），也需要底层代码生成和优化（比如OpenCL是在运行时编译为GPU或CPU代码并发执行的）。而这些需求得以实现，归功于LLVM自身的新前端——Clang。

## 优异的答卷——Clang

前文提到，苹果吸收Chris Lattner的目要比改进GCC代码优化宏大得多——GCC系统庞大而笨重，而苹果大量使用的Objective-C在GCC中优先级很低。此外GCC作为一个纯粹的编译系统，与IDE配合得很差。加之许可证方面的要求，苹果无法使用LLVM继续改进GCC的代码质量。于是，苹果决定从零开始写C、C++、Objective-C语言的前端Clang，完全替代掉GCC。

Clang的加入代表着LLVM真正走向成熟和全能，Chris Lattner以影响他最大的“龙书”封面为灵感，为项目选定了图标——一条张牙舞爪的飞龙。

Clang一个重要的特性是编译快速，占内存少，而代码质量还比GCC来得高。测试结果表明Clang编译Objective-C代码时速度为GCC的3倍，而语法树（AST）内存占用则为被编译源码的1.3倍，而GCC则可以轻易地可以超过10倍。Clang不但编译代码快，对于用户犯下的错误，也能够更准确地给出建议。使用过GCC的读者应该熟悉，GCC给出的错误提示基本都不是给人看的。

比如最简单的：

```
struct foo { int x; }
typedef int bar;
```

如果使用GCC编译，它将告诉你：

```
t.c:3: error: two or more data
types in declaration specifiers
```

但Clang给出的出错提示则显得人性化得多：

```
t.c:1:22: error: expected ';' after
struct
```

甚至，Clang可以根据语境，像拼写检查程序一样地告诉你可能的替代方案。

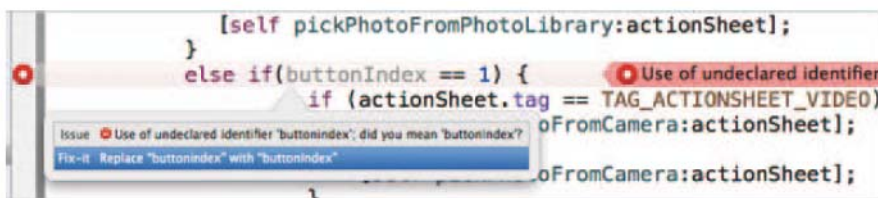
发程序如Xcode、Interface Builder等，皆由Clang编译。到了Mac OS X 10.7，整个系统的代码都由Clang或LLVM-GCC编译。

## LLVM周边工具

由于受到Clang项目的威胁，GCC也不得不软下来，让自己变得稍微模块化一些，推出插件的支持，而LLVM项目则顺水推舟，索性废掉了出道时就一直作为看家本领的LLVM-GCC，改为一个GCC的插件DragonEgg。苹果也于Xcode 4.2彻底抛弃了GCC工具链。

而Clang的一个重要衍生项目，是静态分析工具，能够通过自动分析程序的逻辑，在编译时就找出程序可能的bug。在Mac OS X 10.6时，静态分析被集成进Xcode 3.2，帮助用户查找自己犯下的错误。其中一个功能，是告诉用户内存管理的Bug，比如alloc了一个物件却忘记使用release回收。这已经是一项很可怕的技术，而苹果自己一定使用它来发现并改正Mac OS X整个系统各层面的问题。但许多开发者还不满足——既然你能发现我漏写了release，你为什么不能帮我自动加上呢？于是ARC被集成进Clang，发生了文章开头提到的“开发者们的惊愕”——从来没有人觉得这件事是可以做成的。

除LLVM核心和Clang以外，LLVM还包括一些重要的子项目，比如一个原生支持调试多线程程序的调试器LLDB，和一个C++的标准库libstdc++，这些项目由于是从零重写的，因此要比先前的很多项目站得更高，比如先前GNU、Apache、STLport等C++标准库在设计时，C++0x标准还未公布，所以大多不支持这些新标准或者需要通过一些肮脏的改动才能支持，而libstdc++则原生支持C++0x。而且在现代架构上，这些项目能动用多核把事情处理得更好。



Clang的Xcode环境相比GCC，编译时不但能提供更准确的错误原因，甚至能告诉开发者怎样才正确

不单是苹果，诸多的项目和编程语言都从LLVM里取得了关键性的技术。Haskell语言编译器GHC使用LLVM作为后端，实现了高质量的代码编译。很多动态语言实现也使用LLVM作为运行时的编译工具，较著名的有Google的Unladen Swallow、PyPy，以及MacRuby。例如MacRuby后端改为LLVM后，速度不但有了显著的提高，更是支持Grand Central Dispatch来实现高度的并行运行。由于LLVM高度的模块化，很方便重用其中的组件来作为一个实现的重要组成部分，因此类似的项目会越来越多。

LLVM的成熟也给其他痛恨GCC的开发项目出了一口恶气。其中最重要的，恐怕是以FreeBSD为代表的BSD社区。BSD社区和苹果的联系一向很紧密，而且由于代码相似，很多苹果的技术如Grand Central Dispatch也是最早移植到FreeBSD上。BSD社区很早就找GCC的替代品【注：如[http://en.wikipedia.org/wiki/Portable\\_C\\_Compiler](http://en.wikipedia.org/wiki/Portable_C_Compiler)】，无奈大多都很差（如Portable C Compiler产生的代码质量和gcc不能同日而语）。

一方面是因为不满意GCC的代码品质【注：BSD代码整体要比GNU的高一些，GNU代码永无休止地出现各种严重的安全问题，如<https://blog.delphij.net/2007/01/gpl-gzip1.html>】，更重要的是协议问题。BSD开发者有洁癖的居多，大多都不喜欢GPL代码，尤其是GPL协议第三版发布时，和FreeBSD的协议甚至是冲突的。这也正是为什么FreeBSD中包含的GNU的C++运行库还是2007年以GPLv2发

布的老版本，而不是支持C++0x的但依GPLv3协议发布的新版本。因此，在历时两年的开发后，2012年初发布的FreeBSD 9.0中，Clang被加入到FreeBSD的基础系统。但这只是第一步，因为FreeBSD中依然使用GNU的C++ STL库、C++运行库、GDB调试器、libgcc/libgcc\_s编译库都是和编译相关的重要底层技术，先前全被GNU垄断，而现在LLVM子项目lldb、libstdc++、compiler-rt等项目的出现，使BSD社区有机会向GNU说“不”，因此一个把GNU组件移出FreeBSD的计划被构想出来，并完成了很大一部分。编写过《Cocoa Programming Developer's Handbook》的著名Objective-C牛人David Chisnall也被吸收入FreeBSD开发组完成这个计划的关键部分。预计在FreeBSD 10发布时，将不再包含GNU代码。

LLVM在短短五年内取得的快速发展充分反映了苹果对于产品技术的远见和处理争端的决心和手腕，并一跃成为最领先的开源软件技术。而Chris Lattner在2010年也赢得了他应有的荣誉——Programming Languages Software Award（程序设计语言软件奖）。P

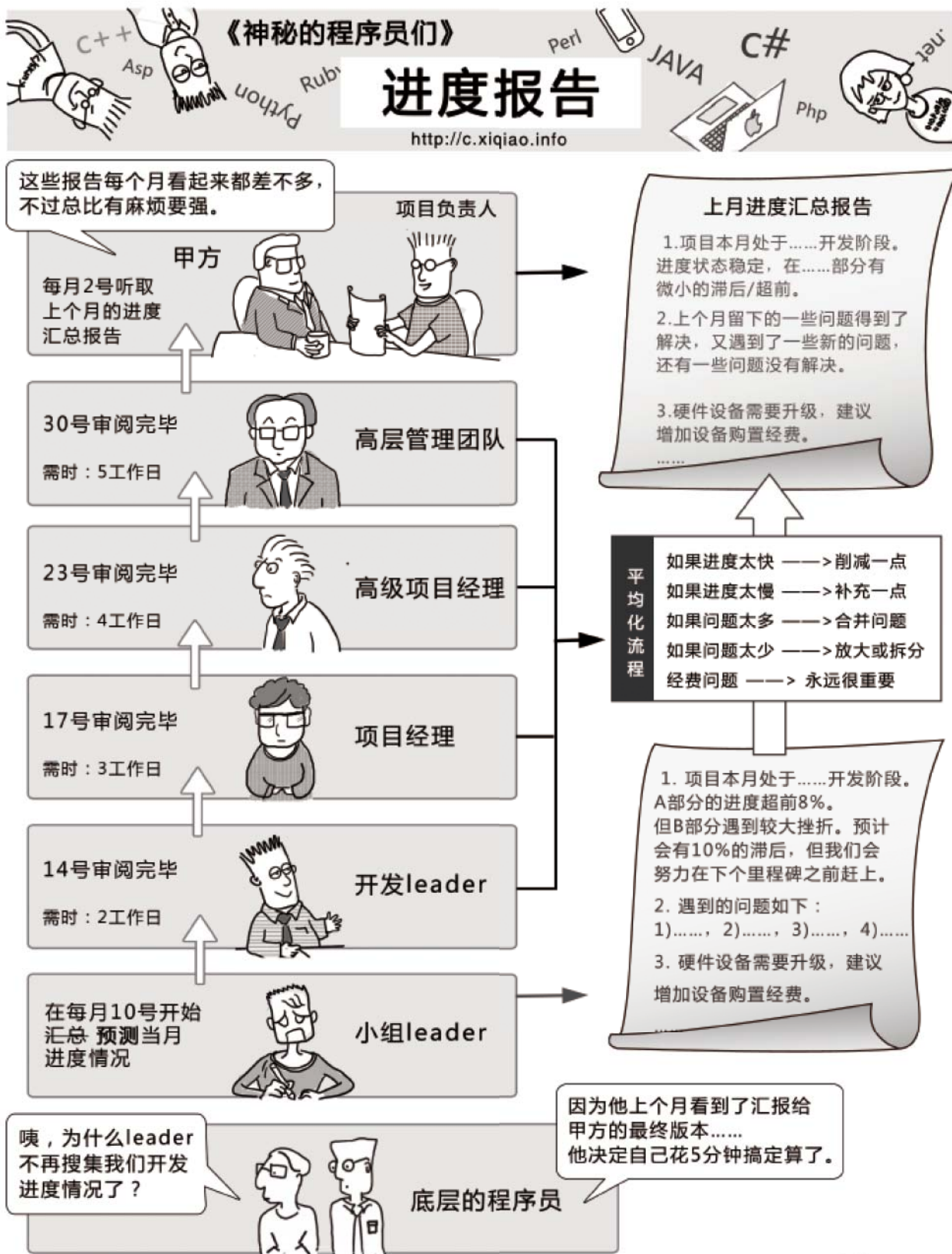


王越

美国宾夕法尼亚大学计算机系研究生，中国著名TeX开发者，非著名OpenFOAM开发者。

责任编辑：白羽中 (balyz@csdn.net)





作者：**西乔**

设计师，项目经理。06年起携创业团队从事Web技术外包开发及产品咨询顾问。  
如果你有什么好玩的关于程序员的故事、对话、代码，愿意通过漫画的形式分享，  
请给西乔发邮件：[arthur369@gmail.com](mailto:arthur369@gmail.com)

